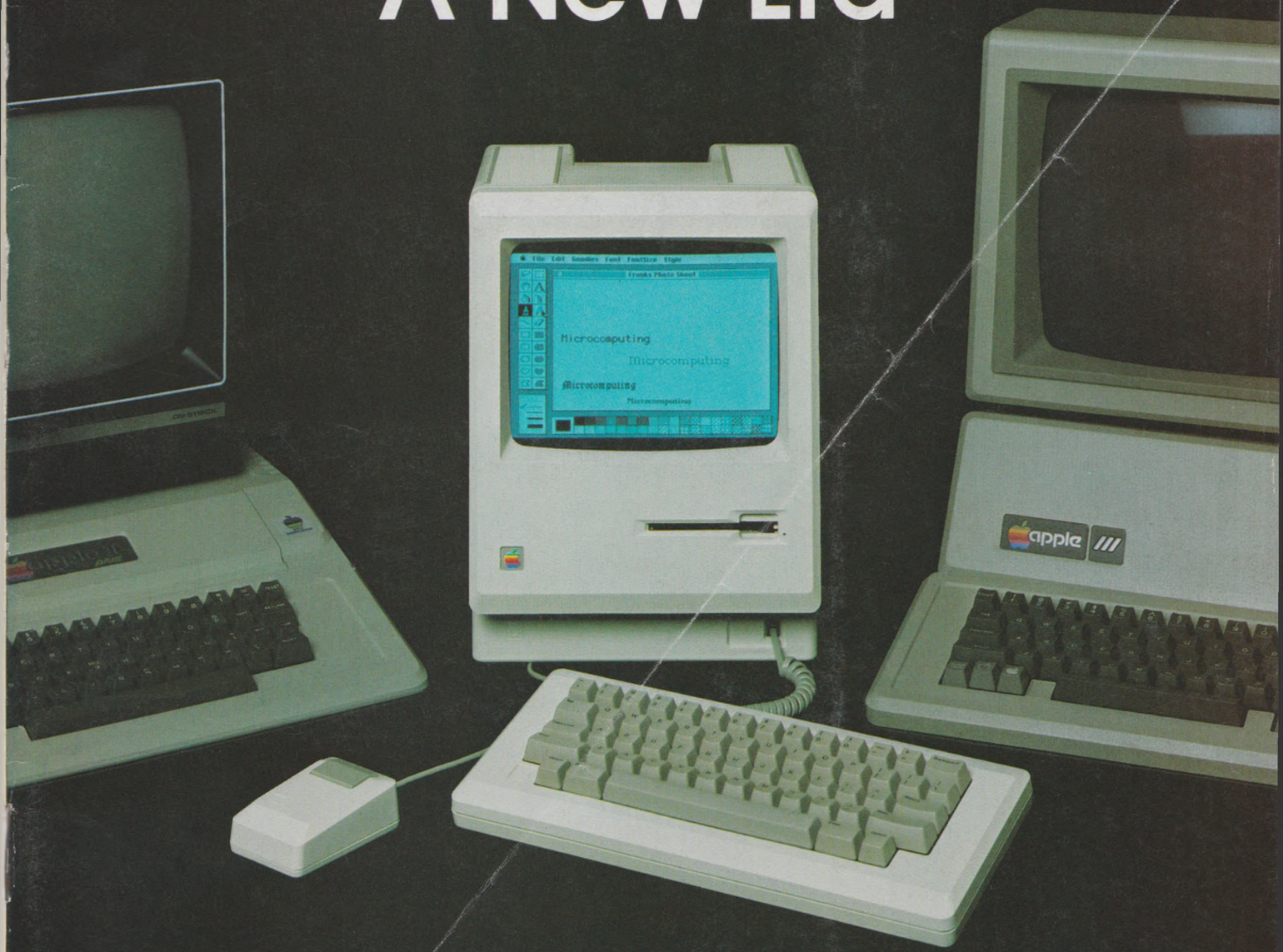
Soup Up Your Kaypro

MICROCOMPUTING®

WAYNE GREEN INC.

March 1984
USA $2.95
Number 87

Macintosh— A New Era



0 74470 65945 03 A

Going the Distance with Telecommunications

The Most Powerful $49.95 Cartridge Word Processor for the Commodore 64™

WRITE NOW!

The *WRITE NOW!* Word Processor provides you with the easiest to use & fastest to learn format available for your C-64. It gives you the features you expect in a professional word processor & some unique features not available in any other C-64 word processor.

With *WRITE NOW!* you have unlimited recall of often used text; global string search makes it easy to find or replace any copy. Editing copy is made easy with simple commands and the full block command enables you to move or delete blocks of copy. You can insert or delete text at will as you scroll up or down through your document line by line, page by page, or you can jump directly to the beginning or the end of your document. *WRITE NOW!* gives you the option of text centering & justification, along with the use of multiple line headers & footers. Page numbering can be set at the top or bottom of a page and either in Arabic or Roman numbers.

Other features of *WRITE NOW!* include:

- INTERFACES with:

  MAIL NOW! Professional mailing list processor for the C-64.

  SPELL NOW! 30,000 word dictionary.

  GRAPH NOW! Full featured graphics and chart processor.
- program on cartridge (time not wasted loading from disk or tape).
- save text to tape or disk.
- sends any 'special' codes to the printer — even in the middle of text (special codes are 'user-definable').
- optional joystick control for scrolling.

Included with *WRITE NOW!* is a 100 plus page easy-to-understand instruction booklet — all for the Manufacturer's Suggested Retail Price of $49.95.

See a complete line of American made Cardco Products at a computer store near you, today.

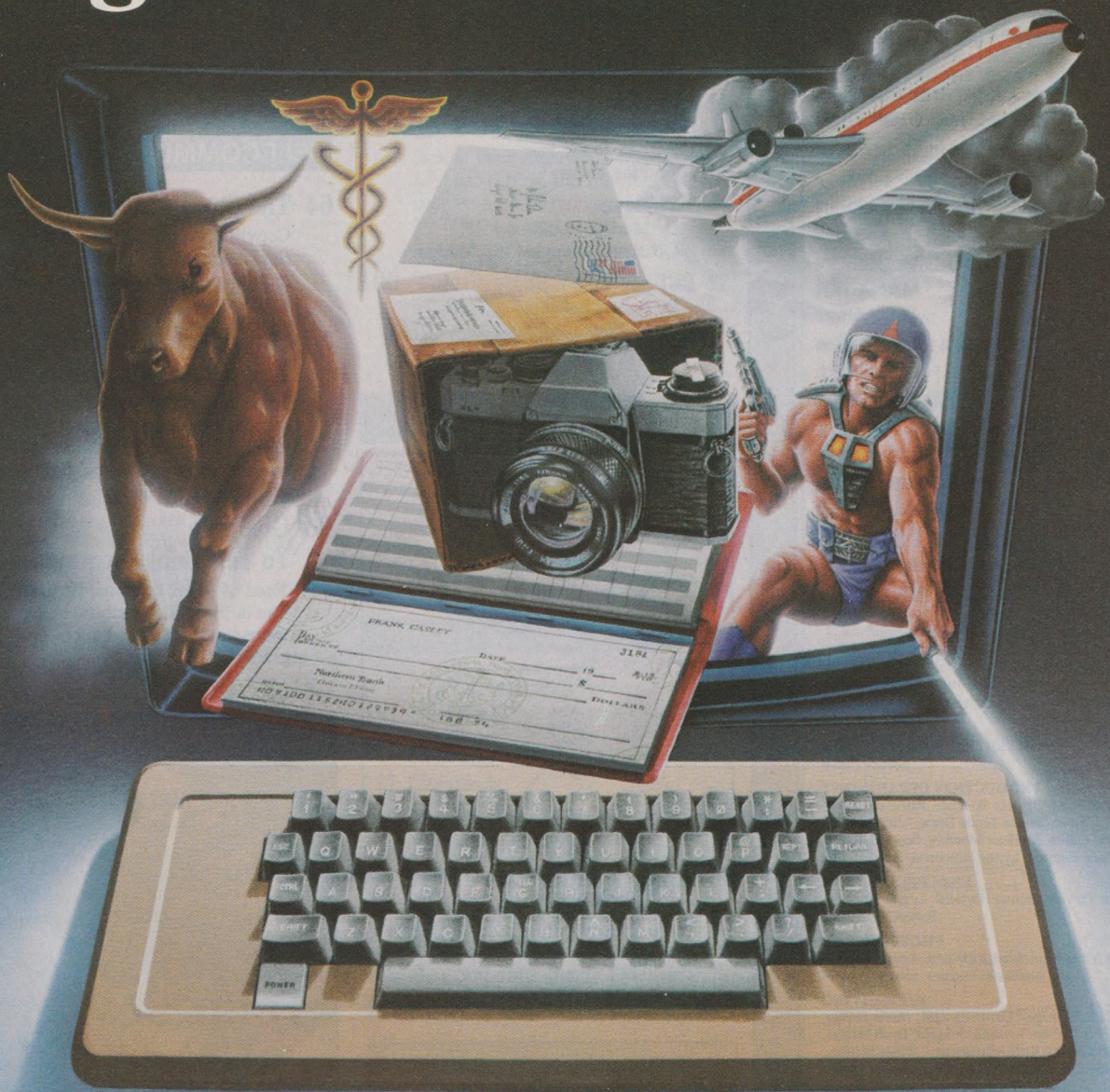
313 Mathewson • Wichita, Kansas 67214 • (316) 267-6525

C-64 is a registered trademark of Commodore International

Circle 369 on Reader Service card.

cardco, inc.

We don't care which computer you own. We'll help you get the most out of it.



CompuServe puts a world of information, communications, and entertainment at your fingertips.

CompuServe is the easy to use videotex service designed for the personal computer user and managed by the communications professionals who provide business information services to over one fourth of the FORTUNE 500 companies.

Subscribers get a wealth of useful, profitable, or just plain interesting information like national news wires, electronic banking and shop at home services, and sophisticated financial data. Plus, a communications network for electronic mail, a bulletin board for selling, swapping, and personal notices and a multichannel CB simulator.

You get games on CompuServe, too. Classic puzzlers, educational, sports and adventure games and fantastic space games featuring MegaWars, the "ultimate computer conflict."

To learn more about CompuServe, call toll-free, 800-848-8199, for an illustrated guide to the CompuServe Information Service. The videotex service for you, no matter which computer you own.

CompuServe

Consumer Information Service, P. O. Box 20212
5000 Arlington Centre Blvd., Columbus, OH 43220

800-848-8199 In Ohio Call 614-457-0802

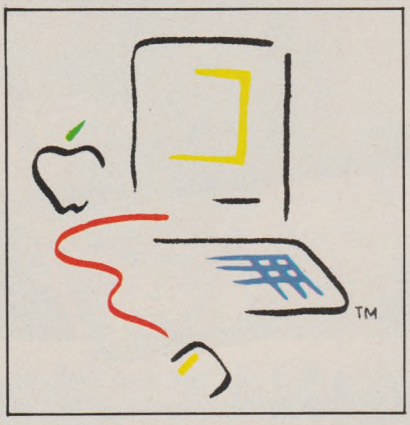
An H&R Block Company

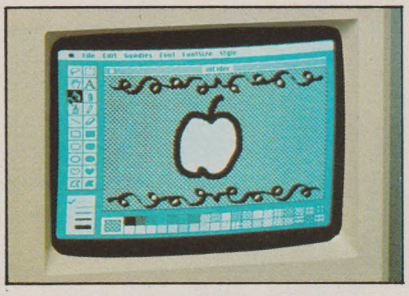
Editorial Offices
80 Pine St.
Peterborough, NH 03458

MICROCOMPUTING

THE COLUMNISTS

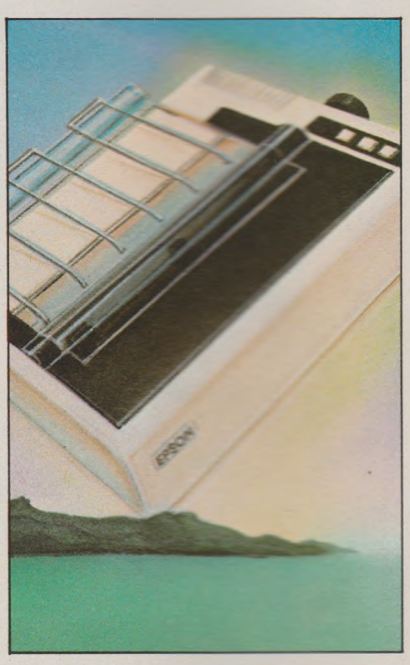
COVER: Macintosh





Mac: A Tempting Apple? Page 66

TELECOMMUNICATIONS



The Epson Also Rises Page 82

Volume VIII No. 3
Contents: March 1984

ARTICLES

DEPARTMENTS

Cover design by Howard Happ.

Cover photo by Frank Cordelle.

Microcomputing (ISSN 0744-4567) is published monthly by Wayne Green, Inc., 80 Pine St., Peterborough NH 03458. U.S. subscription rates $25, one year; $53, three years. Canada and Mexico $27.97, one year, U.S. funds. Foreign $44.97, one year; U.S. funds drawn on U.S. bank. Foreign air mail subscriptions—please inquire. Nationally Distributed by International Circulation Distributors. Second-class postage paid at Peterborough, NH 03458 and at additional mailing offices. Phone: 603-924-9471.

Postmaster: Send form #3579 to *Microcomputing*, Subscription Services, PO Box 997, Farmingdale, NY 11737.

Microcomputing is certified by the Audit Bureau of Circulations.

Microcomputing is a member of the CW Communications/Inc. group, the world's largest publisher of computer-related information. The group publishes 42 computer publications in 18 major countries. Nine million people read one or more of the group's publications each month. Members of the publication group include: Australia: *Australasian Computerworld, Micro Magazine*; Argentina: *Computerworld/Argentina*; Brazil: *DataNews, MicroMundo*; Denmark: *Computerworld/Danmark, MikroData*; France: *Le Monde Informatique*; Germany: *ComputerWoche, MicroComputerWelt, PC Welt*; Italy: *Computerworld Italia*; Japan: *Computerworld Japan*; Mexico: *Computerworld/Mexico*; Norway: *Computerworld Norge, MikroData*; People's Republic of China: *China Computerworld*; Saudi Arabia: *Saudi Computerworld*; Spain: *Computerworld/Espana, MicroSistemas*; Sweden: *Computer-Sweden, MikroDatorn, Min Hemdator*; United Kingdom: *Computer Management, Computer Business Europe*; United States: *Computerworld, HOT CoCo, inCider, InfoWorld, ISO World, Microcomputing, PC World, 80 Micro, RUN.*

By Wayne Green

Pondering Over Portables

The Past, Present And Future Of Mobile Micros

Those Lap Computers

When Osborne came out with his portable computer it was an instant success, there was a collected "aha." People wanted portable computers. The success of the Kaypro and Compaq, improvements on the Osborne 1, solidified this perception. Baloney, thought I, the Osborne 1 is a hit because it is cheap and comes with software that is all ready to go.

In fact, by the time you figured in the cost of that nice software the 1 wasn't just cheap, it was a steal. And people lined up for the steal. George Morrow, one of the smartest engineers in the business, scoffed at the modest 1 guts and at the misperception that portability was the big deal. Cheap was the deal, opined George, so he brought out the Morrow MicroDecision and sales boomed. He put better guts in his unit, but I'll bet most of the MM customers bought on price and didn't until later discover what a bargain they'd stumbled on.

Now I'm not saying that there aren't some people who really want a portable computer—one complete with disks. I just don't think there are all that many. Most Osborne 1 units are, I'll bet, sitting on desks accruing dust underneath, not being carried on planes or taken home at night from work. Did you ever try to carry one of those damned things? Osborne must have gotten a woman wrestler for his television commercials in order to make it look easy to carry that kluge.

The Osborne 1 surprised me from several views. It was obviously a whole lot larger than it had to be—with those slots for disk storage and so on. It was heavier than it had to be. The screen was smaller than it had to be. And the whole thing looked more like a Russian tank radio than a computer. Ugly. His Vixen was a far better looking unit to my eye.

Otrona brought out their Attache computer, showing what could be done in space saving. It had a better monitor and even though it was the same size as the Osborne 1's monitor, it was easy to read for word processing. The keyboard was cleverly labeled to make their word processor a cinch to use—even for me. I get lost between trying to remember Scripsit, Easy Writer, Word Pro and so on. Every word processor is different.

I Had to Have One

The minute I saw the Tandy 100 I had to have one—and it's been with me just about every minute ever since, some eight months as of this writing. I've had it on so many plane trips I can't count them, including two trips around the world—with no fuss from any airline.

Before the 100, I was using the Sony Typecorder. I carted that all over the country and around the world for more than a year before the 100 appeared and bumped it from my briefcase. The Typecorder was a fantastic portable typewriter, able to store up to 100 pages on its built-in microcassette recorder. I wrote letters, articles, editorials, memos and such and sent them back from all over via microcassettes by airmail. No problems.

The Typecorder's LCD showed only half a line of type—40 characters—at one time, so it wasn't all that handy for editing. But it was small, light, fit in the lap just fine and made it possible for me to work in the car, on planes, or wherever I happened to be.

For some weird reason, Sony kept its Typecorder a deep secret. I tried to interest them in doing some promotion, but nothing came of it. I'd estimate that Sony wasted away at least $100 million in sales of that beauty. Then it compounded this enormous marketing gaffe by not keeping up with the technology. For instance, it took more than a year to get word-wrap into the operating system ROM.

The price was a bit steep at $1200, but then the Sony quality made it seem reasonable. It would have been worth double that for the convenience it brought me. They eventually brought the price down to about half that, but it was too late by then.

Sony is usually good about its advertising, but in the case of the Typecorder I saw one ad for it several months before it was available anywhere and that was it for about a year! Sony showed it at the consumer electronics shows, wowing the industry. But my attempts at getting through to anyone at Sony were so I gave up and bought a couple for us to use.

The same story was repeated with the Sony SMC-70 microcomputer. They introduced it to the press in New York and then did just about everything they could to keep the system a secret. One of the biggest features of the 70 was the incredible graphics possibilities, so wouldn't you know it took more than a year before the software for the graphics was released? Hey, Sony, this is a fast-moving industry—it really won't tolerate that kind of pace.

HX-20: Not Long Enough

The next lap-sized unit was the Epson HX-20. The built-in microcassette recorder is great and the microprinter is also a great idea. The four line by 20 character screen was much too small a window. The screen should be at least 40 characters wide, with 80 being optimum. I managed to get an HX-20, but then found it had zero software, so it just sat on my coffee table.

Then Came Tandy

The impact of the Tandy 100 was not lost on the sharper people in the industry. There are millions of businessmen who need a really portable typewriter, calendar, calculator, clock, memo pad, address and phone index and so on. NEC and Olivetti both made deals with Kyocera, the outfit making the 100, so they have almost identical computers, the 8201 and M-10.

An Eye Peeled for Asian Laptops

In October, I made a quick (and I mean quick) round of the computer shows in Europe and Asia. I had my eye peeled for further lap computer developments. The most remarkable of all was the Dulmont, made in Australia. I spotted this at a small microcomputer show in Singapore. It had an 8 × 80 display, but the company promised 16 lines by Comdex time.

The Grid portable was a big hit at NCC in 1982, but the firm stiffed the press and compounded its problems with a high price, so the system never really got started. The Gavilan, introduced at the 1982 Spring Comdex, is still not available. It sure was the hit of that show, but the company's slowness in getting the bugs out and hitting the market may kill them. I made a list of the lap-sized systems I have literature on so far—19 of them—and this is before the Winter Comdex. I'll bet at least 20 more show up there.

What is the market for briefcase computers? I'll bet that if the industry can build them fast enough we'll see more than two million sold in 1984 and maybe five million in 1985.

I guess it is clear that I see the lap computer as a new market, one that won't particularly interfere with desktop computers. These lap units serve a different purpose. I find that I need to have my desktop computer to work in conjunction with my 100. In this way my editorial material can be dumped through the RS-232C connector to disk and then edited directly without ever having to be rekeyboarded.

Letters I want to send to several people can be saved to a disk. In practice, I've found the Leading Edge Prowriter to work beautifully with the 100, giving me fast and fairly good looking dot matrix letters. For governors and senators (and kings) my letters go into the desktop computer and then to a letter quality printer.

We're going to see some fast developments in these lap computers. Sixteen lines by 80 characters will be the next standard, then 24 lines and eventually 32 lines. I think that will be enough. The need for more than RAM storage has to be met.

The HX-20 and Typecorder microcassette is a reasonable approach, but as with all tape systems it is slow. I have to take a Sony Walkman Professional recorder with me to save things when I am on trips and find my memory full. It works, but it's clumsy.

A few of the lap computers are planning on small disk systems. I'm not convinced this is a good approach yet. We'll see. This might be the ideal place for that old Exatron stringy floppy system. That was an invention looking for a good application. I thought it was coming back to life when TI showed it in conjunction with its Model 100 and their 99/2 computers. Unfortunately, neither of these systems ever got much beyond their showings at CES. It's a good technology for this application, with plenty of memory storage, fast search potential and an imperviousness to shock we have to have with lap computers.

Not Worth Its Weight

Since printers really aren't needed most of the time, I'm not sure that it's worth the weight to have them as an integral part of the lap computer. I've been promised a Sharp 5000, which has what is supposed to be a good printer built in, so I'll let you know how that goes.

So far, the printer I like the best is a small one by NEC that prints 40 columns in dot matrix, the PC-2021. It retails for about $100 in Japan. That would do me for notes, for saving things I want to file, short letters on trips, and so on. It's small and light enough to go into the briefcase with the computer. It weighs in at 11¼ ounces, and is 5.2 × 4.4 × 1.6 inches.

Panasonic has a somewhat larger printer, the JR-P20, which sells for $150 in Japan.

It weighs about a pound and runs from 6V, prints on 3½-inch paper, but is 5.75 × 7.75 × 1.5 inches. Nice. Epson's P-40 printer prints 40 or 80 columns on 4½-inch paper. It weighs about 1½ pounds and is about 8.6 × 5.1 × 1.8 inches and runs from built-in nicad batteries.

Speaking of printers, yes—oh yes—I'm familiar with portable typewriters. I've been traveling for quite a few years now and almost always hauled along some sort of portable—like the old Hermes Rocket, the favorite of newspapermen. When the Brother EP-20 came out, I got one from JS&A and took it on a few trips, but by then my Typecorder had rendered the Brother almost useless.

Brother now has the EP-22, which is supposed to be not only a nice portable typewriter, but plug into my computer and act as a printer. I'll have to try one and see if it can really work with the 100. One of the more serious problems with the 100 is the weakness in the word processor that prohibits me from putting in special print codes. I sure would like to be able to tell my printer to advance to the next page without sitting and waiting beside it. I'd also like to be able to use accents and special characters that my printer has, but which are not available.

A Million Dollar Market

What is the market for briefcase computers? I'll bet that if the industry can build them fast enough we'll see more than two million sold in 1984 and maybe five million in 1985.

They are just too valuable for the average businessman to pass up—and at around $1000 less than a good typewriter, why on earth should anyone pass them up? The only negative is that since this type of product is being made by smaller firms, the advertising budgets will be smaller and thus it will take longer to reach businessmen. □

A Micro Renaissance?

Sharpen Your Pencils And Write, Write, Write

On p. 66 of this issue you'll find my review of Apple's newest entry in the personal computer market—the Macintosh. I think and hope that this machine marks the beginning of another generation of personal computers. Finally, here is an affordable machine that will appeal to the person who isn't interested in computers as ends in themselves but, rather, is interested in what a computer can do for him—provided he doesn't have to literally hack through a typical micro's operator's manual.

The Apple Spirit

When I first visited Apple to view Macintosh, I spent time with the people most responsible for the design of Macintosh. I was impressed with their spirit of invention and innovation. They weren't interested in manufacturing just another computer; they were interested in producing a machine that would put a computer into the hands of millions of people who might otherwise never realize how a computer could influence their personal and work lives. Apple, too, is to be lauded for providing an atmosphere that has not only attracted a talented group of people but has given them the framework in which to be creative.

Since the introduction of the IBM PC, manufacturers' creativity has been stunted. Computer makers have deluded themselves into thinking that the market really needed another compatible. At Comdex last November there were no fewer than 20 compatibles introduced from well-known companies as well as obscure new companies. I would like to meet some of their financiers—seems there is this bridge in Brooklyn . . .

Software writers have been a bit more innovative in their wares, but as success breeds success, their efforts have been funneled into programming more word processors, databases and spreadsheets for the PC. Not that we don't need better programs. It's just that so many of them are alike.

An Open Window

Macintosh opens a new window (pun intended) for all of us in the microcomputer business. I hope that vendors will see Macintosh not only as a fantastic opportunity for new enterprise but as a renewed impetus for innovation and creativity. Perhaps we'll see a resurgence of the earlier days when there was genuine excitement over the then new TRS-80 Model I and Apple II. There is a new market waiting to be awakened. Thanks, Apple, for the opportunity.

K.T.

Software Review Board

As we've been alluding to for some time, there are some exciting things in the works for *Microcomputing*. We can't give you any more details yet, but we can discuss one of the changes we'll be making.

You've probably noticed that we print several software reviews each issue. In order to keep our reviews timely and consistent, we are creating the *Microcomputing* Software Review Board. We welcome your applications.

Basically, what we're looking for are readers/reviewers with a good grasp of the microcomputing field in general and specific expertise in one area (word processing, databases, languages, utilities . . .) or with one system (Apple, IBM, Tandy, Heath, CP/M and S-100-based machines). Equally important is a flare for writing.

As we receive software, we'll sort it and send it to the appropriate board member.

If all this sounds interesting, we'd love to hear from you. Send a letter describing your background and a writing sample to Software Review Board, *Microcomputing*, 80 Pine St., Peterborough, NH 03458.

NEVADA SOFTWARE

FOR CP/M®. TOP QUALITY, BOTTOM PRICE.

Nevada makes it easy and economical to get the software you want and need. Need a better BASIC? We've got it. Is BASIC not solving your business or engineering problems? Nevada COBOL or FORTRAN will. The media praises Nevada, and it's priced right: just $39.95—about 1/10 what comparable quality costs. For about half the price of one competitive package, you can own the *entire* Nevada Software library! So don't wait; order one or two or all of our bargains—today.

NEVADA BASIC™

Diskette & 220-page manual
For business, education engineering, science

Finally, a better BASIC. This straightforward language lets beginners write useful programs without limiting them to simple programs. New Nevada BASIC's interpreter has Prof. Starkweather's great built-in full-screen text editor. You can define single- *and* multi-line functions. Plus there are full-matrix operations, Random Access and Sequential files, program execution with a simple command, BCD Math—no round-off errors. With Nevada BASIC, micros can run like minis costing thousands more.

NEVADA COBOL™

Diskette & 165-page manual
For business

Whether you do business computing or learn computing for business, COBOL is *the* language; more business application software is in COBOL than in all other languages combined. Based on ANSI-74 standards, Nevada COBOL offers many advanced features: Random Access and Sequential files, debugging capability, COPY statement, character string, 16 bit binary and decimal data types. Colleges use the fine documentation as classroom texts. It's field-proven by 10,000+ worldwide business, government and education users. Join them.

COBOL Application Package—Book I. Superior user documentation that saves even experienced programmers many hours.

NEVADA FORTRAN™

Diskette & 214-page manual
For science & engineering

"If you want to learn or teach someone FORTRAN, this is the package to buy." ACCESS, *March/April 1983.* For learning and teaching, for scientists and engineers, Nevada is the perfect FORTRAN. Based on ANSI-66 standards (FORTRAN IV), its advanced features include IF...THEN...ELSE constructs, COPY statement, CHAINing with COMMON, TRACE style debugging, and 150 verbal error messages. And you can intermix in-line FORTRAN and Assembly Language statements for special micro needs. Requires 48K RAM. If you're shopping for FORTRAN, look no further.

NEVADA PILOT™

Diskette & 131-page manual
For dialogue

Perfect for training, testing, virtually all programmed instruction, and word puzzles. It's the ideal companion language for BASIC, COBOL, and FORTRAN application packages, because it so quickly solves training and documentation problems. Nevada PILOT meets all PILOT-73 standards and has many new features including a built-in full-screen text editor. Prof. Starkweather's documentation is exceptional; the manual comes with 10 free programs. See *MICROCOMPUTING* review, January 1983, and you'll be convinced.

NEVADA EDIT™

Diskette & 59-page manual
For text editing

"A well-thought-out product with excellent documentation and an astoundingly low price." MICROCOMPUTING, *May 1983.* Now, high-quality text editing for micros. A character-oriented full-screen display editor, Nevada EDIT is great for program editing as it's specifically designed to create COBOL, BASIC, and FORTRAN programs. Simple to configure, you customize tab stops, default file type, keyboard layout, and CRT by menu selection. Nevada EDIT may pay off better than any software purchase you've made.

EACH ONLY $39.95

COBOL Applications Package—Book I: $9.95

Money back guarantee: you must be completely satisfied, or return the packages(s)—in good condition with the sealed diskette(s) unopened—within 30 days, and we'll gladly refund your money.

SINCE 1977

Send to
ELLIS COMPUTING, INC.
3917 Noriega Street
San Francisco, CA 94122 Phone 415/753-0186

Please send me: **Software Packages**
☐ BASIC ☐ COBOL ☐ FORTRAN ☐ PILOT ☐ EDIT

Disk Format
☐ 8" SSSD (Standard CP/M IBM 3740)
5¼" Diskette for:
☐ Access
☐ Apple CP/M
☐ DEC VT 180
☐ Epson QX-10
☐ Heath Hard Sector (Z-89)
☐ Heath Soft Sector (Z-90)
☐ IBM-PC (Baby blue or big blue card)
☐ Kaypro Double Density
☐ Micropolis Mod II (Vector Graphic)
☐ NEC PC 8001
☐ Northstar Double Density
☐ Northstar Single Density
☐ Osborne (Single Density Disk)
☐ Sanyo
☐ Superbrain DD, DOS 3.X (512 byte sec)
☐ Televideo
☐ TRS-80 Model I (Relocated to 4200 hex)
☐ Xerox 820 (Single Density)

Send my order for ____ packages @ $39.95 each Total _____
COBOL Application package @ $9.95 each Total _____
California residents add 6½% Sales Tax. Sales Tax _____
Outside North America, add $6 per package for shipping. (Postage paid within North America.) Checks must be in U.S. dollars and drawn on a U.S. bank. Shipping _____
☐ Check enclosed ☐ Mastercard ☐ VISA **TOTAL** _____

Card # _____ Exp. Date _____
Signature _____
Ship to: Name _____
Street _____
City/State/Zip _____

The CP/M operating system, an 8080, 8085, or Z-80 microprocessor, and 32K RAM are required. Note: Double Density drives can read Single Density diskettes.

Circle 192 on Reader Service card.

Sneak Previews

How do you interact with your microcomputer? Do you still plug away at the keyboard and stare bloodshot at the screen?

Fortunately for your fingers and eyes, there are new ways to get information in and out of a computer. Touch-sensitive screens, voice recognition and synthesis devices and mouses are only a few of the tools used in the new breed of interactive micros.

April's *Microcomputing* looks at this new breed. We'll feature a review of Hewlett-Packard's HP-150, a touch-and-go computer with some fine features. We'll look at Texas Instruments' new Speech Command system, an accessory that lets you talk to your TI Professional. And, so your computer can talk back, we'll review an inexpensive voice synthesizer that attaches to just about any computer. We'll also look at some mouses and some software that uses them.

We're also reviewing a controversial product that takes a subliminal approach to interaction—Expandovision, an accessory that attaches to your computer and TV and flashes messages on the screen to help you quit smoking, enhance your self-esteem, even improve your sex life.

Finally, we'll continue our coverage of Apple's Macintosh, with news and reviews of some of the outstanding new products available for it.

More good things are coming. In May, we'll look at the sometimes-dizzying world of operating systems. CP/M, MS DOS, ProDOS, Unix—we'll tell which does what, and how.

In June, we'll cover the lightweight world of portables, and we'll launch a bunch of new columns that will have something for everyone. In short, our June issue will be a blockbuster. That's about all we can say right now.

A lot is happening in the world of microcomputing and in the world of *Microcomputing*. Join us in the adventure.

A Step Beyond

Times are changing, especially in the microcomputer world, and our payment rates for articles are changing, too. We are, in effect, doubling our payments.

Why? you ask. *Microcomputing* wants the best and we're willing to pay for it.

We're looking for articles written for the advanced computerist—articles that go a step beyond what you'd find in other computer publications.

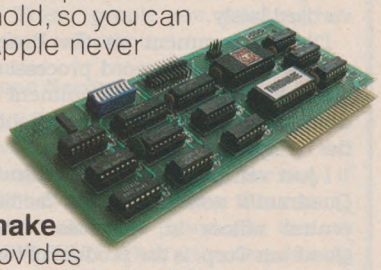
We invite you to send your article or article proposal to *Microcomputing*, Submissions Editor, 80 Pine St., Peterborough, NH 03458. Our brochure, "How to Write for *Microcomputing*," is available upon request.

SEIZE CONTROL.

With FingerPrint™, you've got Apple in the palm of your hand.

FingerPrint functions as the "Print Screen Key" your Apple doesn't have. But that's only the beginning. FingerPrint is more than just a parallel printer interface card. A touch of its button puts any program on hold, so you can make hard copies of *any* graphics or text. And do things Apple never imagined.

FingerPrint consists of the printer interface card connected to a touch-sensitive button that adheres to your keyboard. It also comes with a free printer cable and a disk loaded with programs.



We made FingerPrint smart. You can make it even smarter. The built-in 2K ROM provides functions like **Print Screen, Program Pause, Graphic Dump, Jump to Keyboard or Monitor for De-bugging** and more. With 2K of user RAM, you can invent new functions such as **Keyword Search and Replace** or special tasks only you could dream up.

FingerPrint works with any parallel printer made. And it can be programmed to interface with new printers as they come on the market.

Get the most out of your Apple II, Apple II+ or Apple IIe. Order FingerPrint today. Or visit your nearest dealer for a test flight.

Thirdware has a whole family of innovative computer products. Including a multiple-printer parallel interface, slim-line disk drives, diskettes, disk cases and the only protective sleeves that remain in place while the disk is in use. Write for our latest catalog.

Circle 241 on Reader Service card.

ORDER TOLL-FREE.
1-800-528-6050
Ext. 2112
In Arizona: **1-800-352-0458**
In Alaska & Hawaii: **1-800-528-0470**

THIRDWARE COMPUTER PRODUCTS™

☐ Please send a free Thirdware catalog and complete information on FingerPrint.

☐ Please send ____ FingerPrint(s) @ $149 each. (Add $3.50 shipping & handling for each one ordered.) FL residents add 5% sales tax.

Enclose check or credit card authorization for full amount.

Name _____

Address _____

City _____ State ____ Zip ____

Charge to: MasterCard/Visa (Circle one)

Acct. No. _____ Expires _____

Signature _____

Mail to: **Precision Software, Inc.**
4747 NW 72nd Avenue Miami, FL 33166 MC

By Frank Derfler

Sanyo Makes Its Move

A Hacker's Stroll Down Memory Lane

Welcome

Welcome to "Overview." Each month I use this space to look over the happenings in the microcomputer and data communications industries.

This month, I'll look at a high-tech company that is doing very well and examine the push behind the Sanyo microcomputers that have been so heavily advertised lately.

I'll also comment on the Bruce and James Wordvision word processing program. Finally, I want to comment on the death of microcomputing as a hobby and the birth of hackers in robotics.

I just returned from the dedication of Quadram's new production facility and central offices in Norcross, GA. The Quadram Corp. is the product of two men who saw a world of potential in a limitation the IBM designers built into the PC.

The basic version of the IBM PC was designed with five expansion slots to allow the system to grow. But a relatively basic configuration of the PC uses three of these slots for the video card, disk controller and serial port. Leland Strange and Tim Ferris, two Atlanta businessmen, saw that the combination of slots and expansion cards planned by IBM would never meet the needs of even intermediate-level users of the IBM system.

The first product designed by Ferris and marketed by Strange, the QuadBoard, combined RAM, a serial port, a parallel printer port, and a chronograph on one board.

This combination gives IBM PC owners the ability to get many more functions into their systems. The QuadBoard became a standard part of a great many of the PCs sold. Quadram rode the success of the PC and PC clones like the Compaq.

Contact the author at Box 691 Herndon, VA 22070.

At the end of 1982, Quadram had 35 employees. At the end of 1983, they had more than 300. The company has kept a steady stream of products coming out and plan to be active in the PCjr market. I'll have some words on Quadram's PCjr products as soon as they are available.

A Japanese Challenger

What is Sanyo doing? If you haven't seen the ads for the Sanyo computers, you haven't been reading your newspapers. Sanyo has been plastering newspapers in major cities with ads for its MBC-550 and MBC-555 computer systems. Both the Sanyo marketing approach and its hardware are interesting.

If you believe that the Japanese computer industry is a monolith that takes central direction from MITI, the government organization for marketing high technology, then you might say that Sanyo has been chosen to attack IBM using the ultra low cost approach while Panasonic, Fujitsu, NEC and Mitsubishi attack from other directions.

If you don't believe in the theory of centralized cooperation, then you will probably at least concede that Sanyo is using price as its major weapon to gain a substantial piece of the U.S. microcomputer market.

The Sanyo MBC-550 and MBC-555 are being marketed through an intensive cooperative advertising campaign as IBM PC clones and IBM PC compatibles.

With prices below $1000 and $1500, respectively, the Sanyo MBC-550 and MBC-555 seem to offer a significant price advantage over the IBM PC.

This apparent advantage is increased by the amount of software included in the package. The retail price of the included software is considerably more than the price of the entire Sanyo system.

Sanyo has chosen to market its microcomputers through local business machine and office supply stores. I can vouch for the fact that, at least on the East Coast, the Sanyo systems seem to be available in quantity. Every store-front office supply outlet I visited was ready to put a Sanyo microcomputer into my hands.

Not a Clone

While it may have some degree of DOS and disk drive compatibility with the PC, the Sanyo microcomputer certainly doesn't qualify as an IBM PC clone, as it has been described in some ads.

Both the Sanyo MBC-550 and the MBC-555 sport a low-profile cabinet that has a gray metallic finish.

The separate keyboard has a nice feel, but it only has five special function keys (as compared to the IBM PC's ten) and lacks an Alt key. The cabinet is too small to hold expansion cards designed for the IBM PC bus and has only a limited expansion capability using Sanyo's unique expansion modules. The 550 (selling for between $900 and $995) contains one single-sided disk drive. It can be expanded and dealers are selling 550s with two single-sided drives, 128Kb of memory and a monitor for about $1400.

Now you know where all of those single-sided disk drives from IBM PC upgrades have gone! The 555 (locally selling between $1500 and $1599) has two single-sided drives and 128Kb of RAM. (If that sounds like an upgraded 550, you're right!)

The Sanyo machines come packaged with MS DOS, Basic and the Micropro family of software, including WordStar, CalcStar, InfoStar, Mailmerge and SpellStar. These programs give a Sanyo buyer software that can be useful immediately.

You can't run all of the packaged software designed for the IBM PC on the Sanyo systems. Many programs, such as Lotus 1-2-3, require double-sided disk drives that the Sanyo doesn't have. The

Lotus package and many others make use of all ten of the special function keys on the PC's keyboard. They also use the graphics ability of the PC that is not completely duplicated by the Sanyo.

The fact that the Sanyo MBC-550 and 555 are not 100 percent compatible with the IBM PC may or may not be important to you. These systems do offer a lower cost alternative to other MS DOS microcomputers. It is possible that Sanyo may take a portion of the MS DOS market from other manufacturers and not particularly affect IBM. After all, there are at least six U.S. manufacturers who claim they need only 20 percent of the market to be successful. They all aren't going to get the percentage they hope for.

The Sanyo systems are interesting both because of the value they represent and the marketing strategy used to sell them.

Bruce and James

Price, marketing and performance are three major factors that drove the development of a major new word processor designed for the IBM PC. Wordvision has been released by Bruce & James program publishers. It has been carefully designed to provide full capabilities and be easy to use at the same time. The documentation describing the operation of Wordvision matches or surpasses the best in the industry. The quality of this program is matched by its aggressive marketing and pricing.

Wordvision has a retail price of $75. The program will be marketed through many different outlets, probably including both computer stores and book stores. Obviously, the publishers hope to sell a great number of programs at this price.

I give Wordvision high marks for both performance and ease of use. The developers tried to consider every practical option and to make the features as easy to use as possible. The program makes good use of the special function keys on the PC. These keys are identified by either their standard F1–F10 designations, color codes or special shapes (heart, diamond, spade and club). The program package comes with keycaps or stickers that can be used to identify the keys used in the program.

Wordvision runs on an IBM PC or equivalent system with at least 96Kb of memory, an 80-column display, one disk drive and PC DOS 1.0 or later.

Wordvision can perform all of the standard functions of insertion, deletion, block moves and text adjustment you would expect from a modern word processing program. Some of its unique and even unusual functions include the ability to quickly swap transposed letters, quickly change letters between capital and lowercase and store text phrases for fast and repeated insertion into the text.

> It was a good time to have electronics as a hobby . . . I would feel a loss for those hardware hackers I used to know, except that I have found them again.

The program allows for hard spaces that can't be changed by later reforming of the text, soft hyphens that can be changed if the text is reformed, and flexible justification of the text. The Wordvision keyboard even includes an undo key, which allows you to decide that you really didn't want to make the last change you entered.

The operator's manual for Wordvision makes use of eye-catching drawings, color, boldface type and excellent organization to help you understand how the program works. The manual contains a built-in brace so it sits on your desk ready for use. It has an index and several appendixes with information on how to use the program with various printers and disk operating systems.

The Wordvision package includes a help file that can be called and reviewed while you are working with the word processor. The help program gives you information appropriate to the portion of the word processing program you are working on. If that help screen doesn't tell you what you want to know, you can move up and down the information tree to find the level of detail you need.

I haven't wrung out this program completely, so it still hasn't replaced WordStar on my systems. But if you're looking for a complete and powerful word processor that is easy to use and well-documented, you should give serious consideration to Wordvision. It is a carefully and professionally prepared program and an excellent value.

The Death of the Hobby

I started writing for Wayne Green in 1969. Wayne and I weren't covering microcomputers then, we were having a good time climbing mountains and towers to install radio antennas and talking around the world using amateur radio systems.

Wayne's *73* magazine was (and still is) mandatory reading for anyone who wanted to get his fingers into radio and communications technology. That was a good time to have electronics as a hobby. Wayne and I both discovered microcomputers at about the same time and the first articles I wrote for *Kilobaud* magazine still involved soldering and putting together parts to make something.

The hobby aspect of the microcomputer industry changed as the name of this magazine changed from *Kilobaud* to *Microcomputing* in early 1982. We all became much more focused on the growth of the industry and on commercial products. The early hobbyists turned professional and often successfully marketed their devices, programs or expertise. Now almost no one in the industry even owns a soldering iron and many fear the idea of inserting an expansion board into their own machine.

I Loved Those Days

Things change and that is as it should be, but I still feel a loss for those hardware hackers I used to know. I would feel really sad, except I think I found them again. I believe I have found the next generation of folks who will happily work for days and weeks on a pile of wires, lights and assorted parts and not be concerned with how soon they can have a public stock offering. These people are working on things they call robots.

Everyone Needs a Hero

Many of you probably shared my curiosity when you saw the release of the Hero 1 robot from Heath. (By the way, if you want to be accepted by this crowd, you have to learn to pronounce that as HEE-ROW WUN. That's the way Hero's voice synthesizer pronounces his name.) Hero 1 is an ambitious project and the literature provided by Heath correctly refers to Hero as a learning device. Hero, and all of the other robots I have seen so far, really aren't useful around the house or office.

Hero lacks the sensors to know exactly where he is at all times and he often doesn't have enough accuracy to retrace his steps. His programming has to be done in assembly language and his arm lacks the strength for major work. But none of these limitations detract from his value as a device to teach the concepts of robotics.

The purists insist that a robot is a mobile device with the ability to manipulate things in its environment in the same way a human being does. This pure definition usually includes the condition that the robot have at least one arm that simulates the action of the human arm. I think it is good to have that pure definition, but the people who build machines within those limits are operating in the same mode as someone who builds a microcomputer using plug boards and

punched cards because that's the way the first computers were designed.

At a recent meeting of robotics enthusiasts, I found many people with positive ideas for building machines that are able to do useful work. But those folks who are limiting themselves to the formal definition of a robot always run into one major limitation: power. It is very difficult to get a source of sufficient voltage and current on a mobile system to do useful work. You can use low power integrated circuits for the on-board computer and reduce the power consumed by the sensors, but the power needed to do what humans consider to be "physical labor" will always be in short supply.

The people who stick to the close definition of a robot are fighting the same problems that have stymied the designers of electric automobiles. The system can do only limited work for a limited time. The greatest problem of a mobile electrical device is how to haul around its power source. Systems that rely on batteries are constantly trying to pull themselves up by their own bootstraps and eventually they fail.

Manipulating the Environment

I would like to suggest that there is a good way to combine our existing microcomputers with the aims and goals of robotics, and have a lot of fun doing it. If the aim of robotic devices is to manipulate the environment, there are many ways to get the job done. My favorite concept is the "wired home." The home is a static structure that has defined functions. Inside and around the home, many repetitive and predictable jobs take place. Some of these jobs, such as vacuuming carpets and mowing lawns, take a considerable amount of power to perform. I contend that the jobs around the house are best performed by dividing them between separate actuators and power units under the direction of a central microcomputer system. As one old saying goes, "Many hands lighten the load"; but then, "Too many cooks spoil the broth." We don't need redundant brains, just multiple manipulators.

Let's see if we can spread the word and build a fire under some soldering irons again.

In a wired home, the existing 110 volt ac power line is put to good use. This power source provides almost all of the energy for the actuators and sensors the house needs. Any mobile systems (for vacuuming and mowing) use only battery power when they are performing their jobs. At all other times they are in a dormant mode charging their batteries. The ac power line serves as the signaling medium for most operations and as an antenna for remote control systems. Important jobs, such as security monitoring, fire protection, utility control and machine/human interaction, are controlled by a central microcomputer working through remote sensors and controls.

The wired home may be viewed by some people as domestic automation instead of robotics, but the goals are the same. The wired home takes advantage of the things we have available: power, wiring and centralized computers to do important work. If you are interested in this concept or have done some work in using these remote sensors and actuators with a microcomputer in a domestic or commercial environment, I would like to hear from you. Let's see if we can spread the word and build a fire under some soldering irons again. □

Circle 307 on Reader Service card.

By Thomas Bonoma

This Word Processor Does It All

Some PC Utilities For Your Easter Bunny List

This month I examine Idea Ware's Idea Processor, Ashton-Tate's *Reference Encyclopedia* for the PC, two office automation programs and three graphics programs for your machine. In addition, I'll look at some good utilities you'll want to consider for your Easter Bunny list. You *do* make a list for the Bunny, don't you?

The Idea Processor

This column is being written on the Idea Processor (TIP), a new word processor/filing package/graphics integrator for the IBM PC. Now, what's that?

This word processor coupled with an integrated card-filing, extracting and retreiving system is more than competent and just *slightly* less than excellent.

Furthermore, the card file program can store and display graphics from your spreadsheet, graphics program, doodlepad or any graphics software that doesn't require a cold boot to start. The word processor can also support these graphics capabilities.

The more I work with it, the better I like it. Let's look at the program in pieces, even though you should remember it's all integrated. The word processing program runs in either color or monochrome and supports both monitors, if you have them just like 1-2-3. A document up to 160Kb in length can be edited. There is on-board help, and a wide variety of printers is supported (but not proportional printing, even on the NEC Spinwriter). Only the Epson lookalikes are supported for integrated text and graphics output.

The IP editor is not a what-you-see-is-what-you-get editor, except for boldfacing, underlining and some few other functions. Mostly, it's a combination of substitution characters (such as an up-arrow symbol for a superscript) and printer codes (e.g., >>ce 1 means to center the next line). For proper formatting, a separate print module has to be called separately.

Justification is on-screen but requires manual invocation. I can find no hyphenation help or soft hyphen capability that allows words to be split. Some will regard this as a major omission; however, as an enemy of hyphenated words and given the good interword spacing job the Print module does with output, I think it's fine.

The on-board help facility is a good one, indeed one of the best I've seen. However, there is no facility for saving text and then returning to its editing or construction. This means that you are discouraged from frequent saves during editing, which is a problem.

Scrolling Along

Generally, scrolling is satisfactory, though sometimes keypresses get ahead of the program's ability to keep up with them (for instance, when doing multiple back-spacing in TIP's special push-everything-down-for-block-insert mode—one of two insert modes available). There is, however, no noticeable problem when entering text, and except for what I've noted above, little to no problem with scrolling. TIP also offers certain scrolling commands that aren't given by other editors. An "automatic" one shows line after line (or page after page) automatically when the scroll lock and cursor pad keys are pressed.

Scrolling by word, line and other blocks is supported, but not by sentence or paragraph. TIP offers about the widest set of wild card search/replace characters I've seen; search/replace is fairly fast and efficient.

There are many other features to this program that are too numerous to do more than mention. All 40 IBM function keys are predefined in the program and kept available on the 25th status line for instant use (F1 switches the display from regular to shifted to control to Alt function keys and back again). This is a major advantage over other systems that either take too much room with complicated menus or else don't have them at all.

Ten keyboard macros are defineable with a simple Control-D and any numeral from 0 to 9 call; macros may contain up to 100 characters each, and may be nested, but not saved from session to session.

TIP has automatic footnoting, list numbering, sophisticated triple right/center/left justification header/footer system with autodate, time, file and page macros. It also does automatic indexing (you just type a print control statement in the text, tell the program what file to save the index entries to and it does it—no alphabetization, though). This program does darn near everything except wash the car and put out the cat.

A wide range of print format and special function keys are available for insertion in the text at any point, which allows flexibility in text formatting.

Unlike many word processing systems that approach TIP's flexibility level, TIP doesn't make you remember the format commands. Instead, a formatting menu "pops up" to replace the screen text. You make some choices and the command is automatically inserted into the text.

Other print formatting commands are equally extensive. Indexing, for instance,

Address correspondence to Thomas Bonoma, 45 Drum Hill Rd., Concord, MA 01742.

"There's No Time Wiser Than After Dark."

BEAT THE APRIL FEE INCREASE! BE WISE. SIGN UP NOW AND SAVE!

Circle 8 on Reader Service card.

When you're looking for solid, up-to-the-minute information, the best place to search is BRS/AFTER DARK. The one system designed for serious searchers, AFTER DARK offers you more databases citing more authorities on more subjects than any other popular online information service—at the lowest cost.

For only $6 to $20 per connect hour, including telecommunications charges, you can personally access the same online files used by major reference libraries and corporations worldwide. These files contain the latest information on subjects ranging from business management, mathematics and education to health, psychology, chemistry, family planning, and just about everything in between.

The range of databases available keeps expanding all the time. AFTER DARK now brings you the full text of the **Academic American Encyclopedia,** as well as the **Harvard Business Review.**

And for microcomputer buffs, there's a new online Software Directory listing major software packages.

AFTER DARK's simple, interactive language and straightforward logic take only minutes to master. The system is outstandingly comprehensive, remarkably fast, and above all, fun to use. All you need is your classified BRS password and any dial-up system. Then, every weekday from six in the evening until the early morning hours—round the clock on weekends and holidays—you can summon up a wealth of valuable information.

Use the coupon to sign up for BRS/AFTER DARK before another evening goes by. Seriously, it's like having your own university library online at home.

BRS COLLEAGUE
AFTER DARK

EXPANDED SERVICES: MORE DATABASES, ELECTRONIC MAIL AND SHOP AT HOME. ONE-TIME SUBSCRIPTION FEE ONLY $50 ($75 AFTER APRIL 15, 1984).

( ) Sign me up for BRS/AFTER DARK now! I understand I will be charged a subscription fee of $50 plus connect hour rates as low as $6 per hour, billed to my credit card account with a monthly minimum charge of $12. (My order must be postmarked before midnight 4/15/84.)

( ) Send more information on BRS/AFTER DARK.

Mail to BRS • 1200 RT. 7 • LATHAM, NY 12110 • (518) 783-1161

Name _____

Address _____

City _____ State _____ Zip _____

Charge to MASTERCARD/VISA/AMERICAN EXPRESS
(Circle one)

MI 2/84

Acct. No. _____

Expires _____

Signature _____

is done with this kind of command, as is centering, paragraph indentation (as for quotations) and the like.

Perhaps the most interesting "feather" command, as TIP calls them, is the >>gf command. A graph saved on disk or in the cardfile (where it can be supported with associated text) is inserted into the text file at print time. The graph consumes 36 lines of print on the output, but the program doesn't lose track of its place and does its integrating job well.

All in all, the editor of TIP is a good one. I rank it right up there with WordStar, FinalWord and Microsoft Word.

It is different from The Word in that it is only partially a what-you-see-is-what-you-get editor. Much of the program's formatting abilities depend on printer keycodes and other macros, like automatic counters, that are seen only in final form when the Print module is called from DOS. If that were all it did, it would be worth the purchase price. However, there's more to come.

The Cardfile can be thought of as an unlimited keyed scratchpad or multiple-window area. Each card can handle about 1500 words, which is substantial. Cards are kept in "drawers," which are in turn kept in "cabinets." Cards are searchable with a sophisticated search option using and, not and or abilities plus wild card options. If a drawer doesn't happen to be in the cabinet you're using, you simply "import" it so you can use it. But here's the real trick. Each and every card in the cardfile supports all the features of the word processor, special formats, the 40 functions keys, print commands and so on.

Communication's Straightforward And Simple

Communication (e.g., extract a block from the word processor and file it in a card or vice versa) between the cardfile and the word processor is a two- or three-key operation; it's staightforward and simple. This is one of the strongest features of the program.

You can search, replace, insert—anything you can do in the regular editor, you can do on cardfile cards. You can also store graphs imported from other programs on cards. This last feature deserves a little more explanation because it will be one of TIP's major selling points.

To import a graph from another program (I have tested this function with 1-2-3, Four-Point Graphics, Delta Drawing, VisiTrend, PCcrayon and a number of other graphics programs), you first run TIP's SAVESCRN.EXE program. This utility, like a RAMdisk or clock program, underlays your application program and essentially replaces the PC's PRTScreen program.

While 1-2-3 is displaying your latest graphics masterpiece, for example, you just hit CTRL+PRTSC; a small prompt appears on the screen and asks "Save to what file:".

> Flaws are few and far between. TIP does darn near everything except wash the car and put out the cat.

Give the graph a name, and *voila!*, a screen image is saved to the disk. If you have an Epson printer, TIP can incorporate, display and print this image. Fig. 3 shows a sample of text with an integrated graph printed by TIP's Print program.

Where's the Buggies?

Enough, already. Where are the buggies, the flaws, the problems? Well, they're there, but they're few and far between. TIP's integrated editor, directory and cardfile front-end menu does not, strangely, have the Print option integrated with it. You have to exit to DOS and then type PRINT to get hard copy of your document (on-screen previewing is supported, but the scrolling is just horrible). This is a minor annoyance, but an annoyance nonetheless. In the copy of TIP I received, the Print program did not work as advertised. That is, if called from the monochrome screen, it would neither switch to the color one to display graphs nor print those graphs on the printer. If called from the color screen, it would do both but would get a lot of "snow" in the text after displaying the picture. A call to the company brought both a revised Print program that fixed the problem and the information that a more extensive set of print graphics drivers are being developed for the IDS Prism printer to allow the Print program to dump graphs in color in your text.

Another problem occurs if you're editing a 60Kb file and you mark a block near the beginning of the text then move to the end and do a block cancel. The program doesn't like this a bit, though I couldn't make it lose data. And there is no mail merge or speller, though TIP works with commercially available spelling utilities.

TIP comes with a disk full of tutorials, a self-running demonstration and a good manual; however, I found it difficult to locate things in the manual when I needed them. Generally, though, everything you need to start up the computer and look like a professional writer in a couple of hours is included. There is even a section at the back of the manual that explains how to work with DOS. I recommend this program.

Ashton's A-Z

I guess the IBM PC has officially "arrived"—it's got its own encyclopedia written about it! Gary and Karen Phillips have edited Edition 2 (where was I when Edition 1 came out?) of *The Reference Encyclopedia for the IBM Personal Computer*, a 700-plus page, two volume looseleaf tome covering all facets of the PC from ASCII to Zork. Zork? That's right; software and hardware vendors' products are included (usually uncritically and with no advertising). Want to know what Blingspatz is? "You could look it up," as Yogi Berra would say.

Perhaps more interesting and useful than minidescriptions of what Lotus 1-2-3 does or how to hit the Help key in SuperCalc are REIBMPC's minitutorials on the Basic compiler, batch files, dBase, the virtues of DOS 1.1 to 2.0, and WordStar customization patches.

While these tutorials vary greatly in quality, as will any compendium of reviews and opinions with more than 12 people contributing, they are generally worthwhile. To take the WordStar entry as a nonrandom example, there are a couple of paragraphs on "WordStar in 15 minutes," and a valuable listing of patches that can be made to customize WordStar (originally published in one of the IBM PC magazines). On the other hand, the minitutorial on Lotus 1-2-3 is just a program description and references nothing about the program's operation.

Most useful and useable, though, are some lists you just can't find elsewhere, at least not in one place. A cross-listing of all statements by function, for instance, is nearly worth the price of the encyclopedia. There is a lucid description of how to make and use .BAT files, a thorough explanation of control keys on the PC and a nice piece on DOS 2.0.

The Bottom Line?

If they update it frequently enough, the vendor information will be useful; otherwise forget that part. The minitutorials are variable at best; some are great, some worthless. But, the real reference material, and at least 20 percent of the 700 pages is devoted to that, is worth the price of admission.

Form Writer

Is there life without heavy advertising? Can good integrated office software be written in Basic? What's going on with some of the lesser-known vendors out there who are writing productivity software? Let's look. Our two candidates are Business Development International's Form Writer (FW) and Harris Micro Computers' Paperwork (PW).

The stated pupose of FW, and the implied purpose of PW, is to provide a degree of office automation to routine writing/filing tasks in a single program.

Form Writer is well-designed for first-time users who want simplicity with functionality.

Form Writer costs $275 and is a set of Basic programs that compose a form writer (letter writing editor); a ten-field, 40-character-per-field database manager; and several useful utility routines.

The programs run in uncompiled Basic (revision 1.1 of FW), and thus are limited by that interpreter's slowness and waits for program chaining between modules. Yet, except in one area we'll come to below, FW makes the best of this situation. Often, you don't know you're working in Basic, a tribute to the program design.

Form Writer has a high degree of integration and is well-suited to novice operators or managers who are more concerned with getting the job done than with program elegance.

When the program is called up, the database manager can be entered to save records. That is, there are ten fields with program-suggested default categories (name, address, and so on) that you may accept or change to suit your needs. Each field title may be 20 characters in length and may contain spaces—a nice touch. The program then allows defaults to be entered for each response field, so that you can preprogram "Mr.," for instance, to start each name field. Only one index key is permitted—that must be field one, which additionally must start with a capital letter. Most users will employ this field for last names of clients, customers or parts, so this is not a problem.

While there is a provision for adding, changing or deleting records as well as doing disk back up and changing disks, the heart of the DBMS system provided with FW is the batch file, which is a named subset (the same liberal naming conventions are used) of the original database selected according to user-provided search specifications. Multiple ANDed and ORed search specifications may be entered, and records additionally may be excluded (undesirable) on the basis of search requirements. The search routines, at least with a small database, are fast and error-free. All search routines, and indeed, all FW choices, are made from the PC function keys.

Once a batch file has been created, you can enter the Form Writer routine itself, a text editor that accepts fields from the DBMS and inserts them wherever instructed in letters, invoices and the like. Though somewhat slow, especially on backspacing (this is a problem with uncompiled Basic code in general), the editor is flexible regarding page/print formats (e.g., one to 20 lines between each line of text, characters per inch, horizontal motion index, margins, even sheet feeders are supported). The editor provides a full-functioned set of commands (line centering, justification, a "style bar" at the side of the page that shows line formatting and visible paragraph markers). While FW is not a what-you-see-is-what-you-get editor by any means, the style bar does show right justification, indentation and the like. Some of these functions are reflected on the screen itself (i.e., centered text is centered, bolded text appears bold and so on). However, the major use of the FW system is to insert, or inject, one or more of the fields from the DBMS into the letter or form being created, such that a print run prints separate letters for each item in the "batch file," thus giving a mail merge capability. To its credit, FW has a fairly sophisticated set of printer configuration files prepared for users with special capability printers; it comes preconfigured for the Diablo 630.

What You Get

What you get with FW is a good amount of DBMS simplicity, function-key menuing and a forms writer that is tolerable for short documents but certainly not suited to tasks more complex than its original purpose, which is mailmerging letters (or other short documents) from the DBMS. These tasks are handled admirably, but the editor subsystem is more useful for single-page rather than multiple-page letters because there is no provision for headers and footers; cursor movement is limited to end/begin file and word left/right, in addition to normal cursor pad manipulations.

Basic's limitations can make back spacing for an entire line seem as if it takes a month. The DBMS doesn't suffer from problems of slowness nearly as much as the editor does.

The overall system is well-designed for the first-time users who want simplicity with functionality. However, the DBMS is stronger in this regard than the editor. Comparatively, FW stacks up only marginally against other integrated writing/DBMS systems. pfs:Write, for instance, is a much better character editor (though it doesn't offer nearly the page/print formatting features of FW) and includes simple mail merge capabilities when coupled with the pfs:File program as well. For the price of FW, you could buy Write and its associated DBMS program. The latter are more flexible, but not as well-integrated as FW.

BDI has supplied a clear and lucid manual for its product, which includes a tutorial but not on-line help.

A machine language revision of FW is supposedly in the works, which certainly

will remove some of my objections to the editor's speed. Let's hope it also incorporates improved error-handling routines.

Paperwork

I was tough on FW, a good program, and I'm going to be a roaring terror on Paperwork. This series of compiled programs also asserts it is a word processing and mailing list program that can handle letter/document, envelope, label and phone/address list production. Why do I say "asserts"? Because I don't know, that's why, and I have the software and manuals in front of me!

It's Not Me

The first time I started looking at PW I put it away in frustration, thinking I was tired and probably couldn't understand the documentation. The second time, *this* time, I know it's not me.

The PW Reference Manual is just that: a reference of commands without an explanation on the program's utility on one hand, and vacuous sections about "talking to your computer" on the other.

The net result is that what the program does is never explained well, and how to use it is not explained either!

Well, let's try manual 2; almost as thick as manual 1, and labeled Version 1.42 Manual Addendum.

Awesomely Unreadable

This little baby explains all the changes that have been made to version 2.4 of PW and it does so in Epson-like 132-column condensed type (awesomely unreadable!). It assumes you already know how to use PW and apologizes for shortcomings in the old version while crowing about changes in this one.

You tell me what the manual says . . . I can't even figure out what the program does! From what I can gather, the program seems to do many of the same things as FW, but in compiled language. It is command driven, and seems to have extensive search and retrieve capabilities in the DBMS but is restricted by a line-oriented text editor. Other than that, your guess is as good as mine. If there were on-board help the user might stand a chance without manuals. As it is, he doesn't.

Three Graphics Programs

Let's take a look at three interesting graphics programs: Delta Drawing, Four-Point Graphics, and PCcrayon with Business Graphics.

Delta Drawing

Delta Drawing is another of Spinnaker's fine educational and learning programs for the IBM PC. It's characteristic of truly classic software; it's as simple or as complex as your stage of development and familiarity with the software allows.

I'm going to be a roaring terror on Paperwork. You tell me what the manual says . . . I can't even figure out what the program does!

A Picture's Worth 1000 . . .

Since pictures are more useful than words when talking about graphics design programs, Fig. 1 shows a sample of DD's output.

In many ways, DD is reminiscent of drawing with the Logo language. That is, there is a "turtle," called a "delta" by DD, that is moved about the screen to form images. To move in any direction, you just type the number of steps and the direction wanted. "8D" moves down eight steps, for instance; to turn, right or left turn commands are issued. Since wraparound is always on, DD doesn't produce the out-of-boundary errors that make drawing troublesome for younger users.

Erasing, filling screen objects with color and putting text on the screen are simple; usually, just a single-key or control-key combination gives access to the function desired.

DD is unique in that it records automatically most of the commands typed as they are employed to make drawings into "programs." These programs are accessed through the PC's ten function keys. They can be edited from a special symbolic command screen available within DD and are recursive in nature. That is, if F1 is defined as a circle and F2 as a left turn plus F1's program, then a program in F3 that says 12F2 will draw a fancy geometric spirolateral design on the screen.

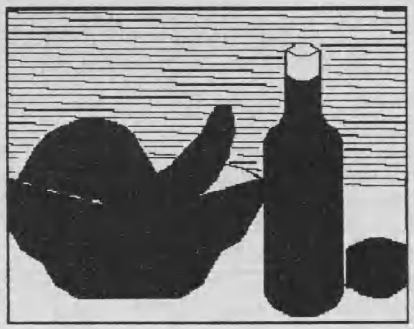


Fig. 1. An example of Delta Drawing's artistic output.

Four foreground colors, two levels of intensity, eight background colors and an add-yellow (switches palettes) function are readily available with either one or two key-code combinations, allowing the production of vivid screen drawings with ease. Additionally, a kaleidoscope mode is available, which essentially puts four delta pointers on the screen instead of one to quadruple the effects of any drawing command issued.

Sizing It Up, Or Down

If all this isn't enough, the results of any recorded program (to draw a house, for example) can be enlarged, reduced, squeezed or stretched out to any of 64 different magnitudes just by adding a command to the program. There is also a random command modifier, which executes the next command issued a random number of times each time it is called, producing a different drawing each time a program is run.

The program has an accumulator function in its graphics language that allows, for instance, the production of increasing sized squares just by drawing one square, adding to the accumulator and drawing the next (larger) one. Finally, there is a reverse and mirror command; the former modifier causes a program or any other command to be executed backwards; the latter lets you draw half of a symmetrical figure and knows how to finish the other half. The program comes with ten sample disk files ranging from simple to very complex drawings, all expressed in its program language.

A disk utility allows file saving, loading, deletions, drive-switching and even a binary file capture to save pictures instead of programs. All of these options are single-key selectable at any point in the program and work flawlessly. If you have either an Epson (GrafTrax), IBM or IDS (460, 560 or Prism) printer, the program will dump any screen or stored drawing in one of two sizes. On the IDS Prism, text will be in the approximate colors used on the screen.

The manual includes a good reference section and a passable, but somewhat weak, tutorial that uses the files stored on disk in addition to teaching you to create your own programs and drawings. All of this is packaged in a rigid case. The software claims to be useful for ages four to adult.

Not Kidstuff

Since I can find no fault with the program, let me comment on this last recommendation. In my opinion, DD is unsuitable for ages four, five, and quite possibly, everyone up to eight years of age. This is not because the young cannot learn to use the "delta" to make simple lines; they can. It is because the key-codes required for even simple changes, like color ones, for instance, are beyond the memory of the very young unless you are willing to

directly supervise them at the drawing task. As a matter of fact, I'd say DD is most useful with the upper end of the age range. DD is a complicated program if you choose to use it to its fullest; it is most definitely not a trivial implementation.

Four-Point Graphics

Four-Point Graphics (4PTG) is another impressive program that has more applications as a design editor/program than as a business or presentation development tool. Like DD, this program is well-designed and supported by a good manual, reference card and other materials.

4PTG is aptly named; you can draw with the cursor in single-point, two-point, three-point or four-point modes. The first is the familiar drawing cursor, supported by cursor-positioning routines, an automove function for long lines with autospeed adjustment and an excellent status line that shows you the drawing color; whether the draw mode is on; what direction was last chosen for cursor movement; which of 4PTG's four display modes are active (see below); and how much memory is available.

Color choices are controlled by the F1 and F2 keys; the palette by F3 and F4 and the background by F5 and F6—a sensible arrangement.

There are facilities in the program for circle drawing of any aspect (squashedness) and size; for automatic circle drawing to produce spirolateral-type drawings; for color fill, wash and even image texturing (using two or more colors in the production of a line or object). 4PTG also allows any two images drawn to be saved in memory and recalled. It has a text draw facility to put characters on the screen as well (38 characters maximum per line). Thoughtfully, there are commands to turn both the cursor and status lines off for screen photography.

Two- and three-point operation are similar to the single-point mode, except the two-point mode is especially useful for drawing lines and the three-point mode is useful for arcs, ellipses and other curves.

But it's the four-point mode that is most interesting: in this mode the four cursors can be used to identify particular images on the screen for saving in the buffers and for image movement, enlargement, reduction, inversion or rotation. The four-point mode is also great for boxes, and it has a facility to texture these with two or more colors as well. Fig. 2 shows some output with 4PTG, produced on an Epson printer.

Four-Point Graphics has a zoom mode as well. In this mode, you can zero in on any small area of the screen and enlarge it for fine editing. Individual control of each pixel in the drawing is possible in this mode.

Don't Let Your Drawings Slide By

In addition to all these drawing commands, 4PTG can create and run demo files, which is to say, a slide show of saved drawings. In this mode, you can control the timing of each screen picture, whether the cursor or the status line is displayed.

Output can be sent to disk, to an Epson (GrafTrax) printer or to a Sweet-P plotter. I couldn't check the plotter routines, but found the Epson print utility to be a flexible way of getting screen output to hard copy. The size of the printed picture is adjustable both vertically and horizontally in an 8 × 8 matrix, giving 64 different printed output sizes. Additionally, the program allows color substitution of any old color for any new one, and it differentiates among colors exceptionally well on the black-and-white printed output by firing only some pins for one color and more for another, giving a shaded appearance to parts of the drawing. Which printhead pins fire for each color is completely under your control. Unlike many print utilities, 4PTG's gives a single-key disk directory option in case you forget what is on the disk.

4PTG's user's manual and reference card deserve special commentary. Though no tutorial is provided and the program has no on-line help, the manual is one of the best I've seen for teaching the program's ins and outs quickly and painlessly. The reference card makes it easy to figure out which command performs what function after the manual has been read once—a fine program for graphics design work on the PC.

PCcrayon

PCcrayon is in many ways the most suitable of the three packages reviewed here for business applications, but it yet is inadequate for serious use in some respects and perhaps better suited to occasional applications.

Fig. 2. Sample output from 4-Point Graphics design editor.

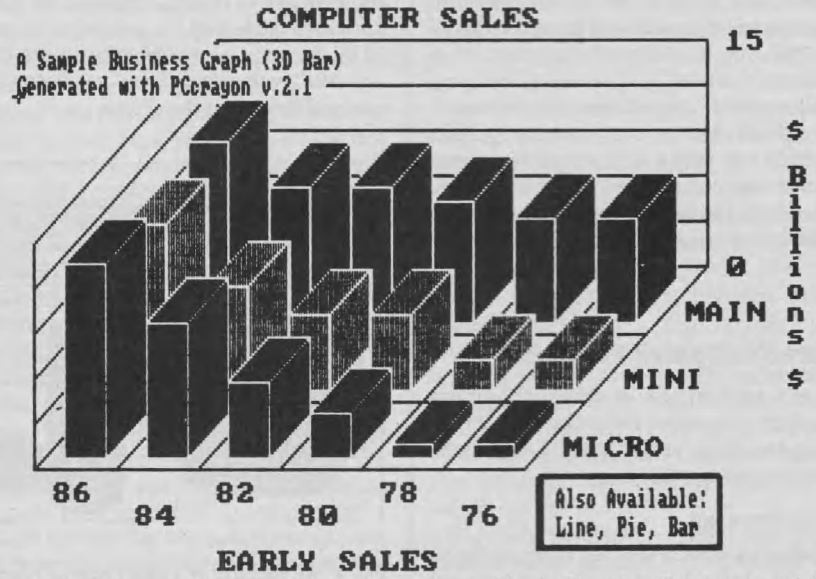


Fig. 3. An integrated text/graphics spread created with The Idea Processor and PCcrayon.

Written in uncompiled Basic, PCcrayon with Business Graphics is in many respects identical to the PCcrayon program reviewed here about a year ago, with the addition of some routines to help produce a variety of business graphs and charts. Fig. 3 shows one of these, printed on an Epson printer.

The basic PCcrayon package is a competent but not terribly easy-to-use graphics editor with slide-show features. The program has good cursor controls, a nice feature for moving text or graphics around on the screen and two built-in sets of commands to construct and manipulate graphics. For instance, arcs, lines, circles, text and other symbols can be placed on the screen in any of nine widths, four colors and two palettes. There are facilities to not only move text and graphics, but to "float" them in animation fashion as well. Additionally, any of a number of symbols (vectors, characters or graphics) can be inserted from the ancillary files accompanying PCcrayon into your drawing. The new revision, for instance, includes a set of freehand-like script letters and a set of flowchart symbols that can be inserted at will.

A second set of PCcrayon commands, called Z commands because they require typing a Z character before they can be entered, is available. Here the screen resolution can be changed from medium to high (the only one of the packages that allows high-resolution black-and-white drawings); commands, whether prompted or not, can be selected; "scenarios," or the creation of drawing steps, can be initiated, ended, saved and replayed with pauses; the screen can be printed (PCcrayon can take up to two minutes to do so); and other functions can be engaged, like drawing a screen border.

It is in this last set of commands that PCcrayon has been enhanced to allow some of the more common business graphics functions to be engaged. Typing Z-GDATA, for instance, brings up a set of data-entry screens on which up to 12 observations for up to three sets of data may be entered, displayed on the graphics screen and edited like any other PCcrayon drawing. Available drawing types are bar, line, pie and 3D bar graphs (vertical only). Additionally, the Z-GRAPH command lets you graph any subset of the three data sets, or all of them together. The 3D bar graph format is especially impressive, but cannot handle negative numbers well.

PCcrayon is supplied with a data disk on which a number of sample files have been stored. The manual is terse at best, and is not helped by the inclusion of a poorly reproduced addendum that describes the latest additions to version 2.1 and gives only a sketchy description of how to create business graphics. The version I received tripped over its own copy-protection scheme and claimed that the program master sent to me by PCSoftware was a bootleg copy; however, because the program was written in Basic, it was possible to bypass the protection scheme.

> Imagine that mad fellow given a heavy dose of hallucinogenics—you'll have a picture of what he did when set loose on the FX-80.

PCcrayon could be helped greatly by the addition of on-board command listings and a help screen, as well as by some more thought as to the ease of interfacing the provided symbol and other files with the main program. The problem isn't that these are hard to access; rather, it is just hard to remember what there is to be accessed, even with an available directory function. PCcrayon is an interesting and versatile graphics system; now it has some limited capabilities for business graphics generation. A new manual, some of the improvements I have suggested above and further development work will make it a full all-around graphics tool for hobbyists and business people alike.

MultiRAM Utilities

MultiRAM is a RAM resident utility for your PC, like a RAMdisk, that gives your properly equipped machine the ability to divide itself into between two and nine phantom computers.

MultiRAM lets you change partitions in memory so that you can run a completely different application, like a DBMS program, on the monochrome monitor while processing words on the color one. Two applications can't run simultaneously, as in Concurrent CP/M, but you can switch back and forth between them with ease; you can even transfer data across these partitions (to a maximum of just over 3000 bytes) without any difficulty.

MultiRAM isn't dependent on having two screens. It's just as happy putting your separate applications on a single monitor, keeping track of each disk used in each application (by way of volume names; this can be a problem with write-protected disks) so you don't lose a whole series of data files by getting confused between partitions, drives and applications. If you have enough memory in your machine, you can, in theory, configure a 256Kb phantom machine for 1-2-3, a 128Kb one for WordStar and still have 180Kb left over for a RAMdisk application. MultiRAM allows you to use other utilities like RAMdisk with no problem and can support partitioning of up to 640Kb of main memory to give you up to nine PCs, eight of which can be directed toward either the monochrome or graphics monitor (assuming you have both) at will.

Changing Your Color

If you do have a graphics monitor, MultiRAM allows you to change the foreground, background and border colors at will, independent of application or whether or not you're in Basic. And, with a monochrome monitor, you can turn character attributes like blinking on or off.

The usual problems with utilities like MultiRAM include an inability to run much software together because of clashing memory calls or other incompatibilities, the inability to get data from one application to another, and uncertainty about use. MultiRAM handles two of these problems pretty well, and goes a long way toward helping with the third.

I found no problems with a variety of software, including Microsoft Word, running as one application in a large partition while I used various Basic programs in a second, smaller partition. Otherwise, I had only infrequent problems with partition-program incompatibility, but these did arise. For instance, The Idea Processor, reviewed elsewhere in this column, did not allow the second partition to operate when it was loaded into the first. Several game programs just won't work as well under MultiRAM. And pfs:File won't work under MultiRAM, at least not with my machine configuration.

MultiRAM's facility for transferring data from one partition to another by just the pushing the Alt and plus or minus keys is a major achievement, and I had no troubles with data transfer. The process is simple; just recall whatever data (character only) you want to transfer to the screen in one partition, mark it by pushing a key, then switch partitions and deposit the data in the new application.

The final area of concern for multiple-partitioning programs is inherent to a degree in all of them. The normal confusion when dealing with up to nine "little PCs" is helped by MultiRAM's on-screen status messages, insurance devices like asking you to replace named disks before you switch partitions, an on-board help facility and a clear, short manual that explains without patronizing and teaches without putting you to sleep or requiring you to learn some new techno-babble.

This is a good program. It and its ilk will be put out of business in a year or so by *true* windowing (or tiling) on a single screen and real concurrency in application; but for now, MultiRAM is the state of the art.

Set FX

I have only one thing to say about this utility: if you have an Epson FX-80, buy it; if not, quit reading. The MX-80 was clearly

developed by some Japanese fellow gone mad with control codes. Imagine that mad fellow given a heavy dose of a strong hallucinogenics, and you'll have a picture of what he did when set loose on the FX-80. This printer has more codes for proportional printing, fancy stuff, quad-strike bit-mapped dumps and so on than you can shake a stick at. Figuring it out, however, is impossible unless you're going to attack the printer in the way you might enemy code in World War II.

Set FX works as advertised to give you different character sets for the Epson (didn't know your FX did that, did you?), including one compatible with all the PC keys, instant setting of the printer to condensed, italics, proportional and 50 other modes—even a graphics character editor so you can create your own typefaces (several are included, including a weird one that looks like what's on the bottom of your checks, a strained version of OCR A).

Like a Porshe Without Wheels

For 60 bucks, it's just hard to go wrong on this program. They show a key in their ads with a note stating that an FX without this program is like a Porsche without a key. A better tag line would have been like a Porche without two wheels; this program should be given by Epson to anyone buying an FX-80.□

The "Big Blue" Black Book

Prices and Addresses of This Month's Featured Products and Companies.

Delta Drawing ($49.95)
Spinnaker Software Corp.
215 First Street
Cambridge, MA 02142

Four-Point Graphics ($415)
IMSI
633 Fifth Avenue
San Rafael, CA 94901

Multi-RAM (requires PC XT, 128Kb, DOS 1.1 or 2.0, one disk, TV or monitor) ($99)
Softsmith, Inc.
2935 Whipple Road
Union City, CA 94587

PCcrayon version 2.1 (with business graphics option) ($125)
PC Software
4155 Cleveland Ave.
San Diego, CA 92103

Paperwork (1.42) ($70)
Harris Micro Computers, Inc.
Software Division
3750 S. Maple Grove Road
Boise, ID 83709

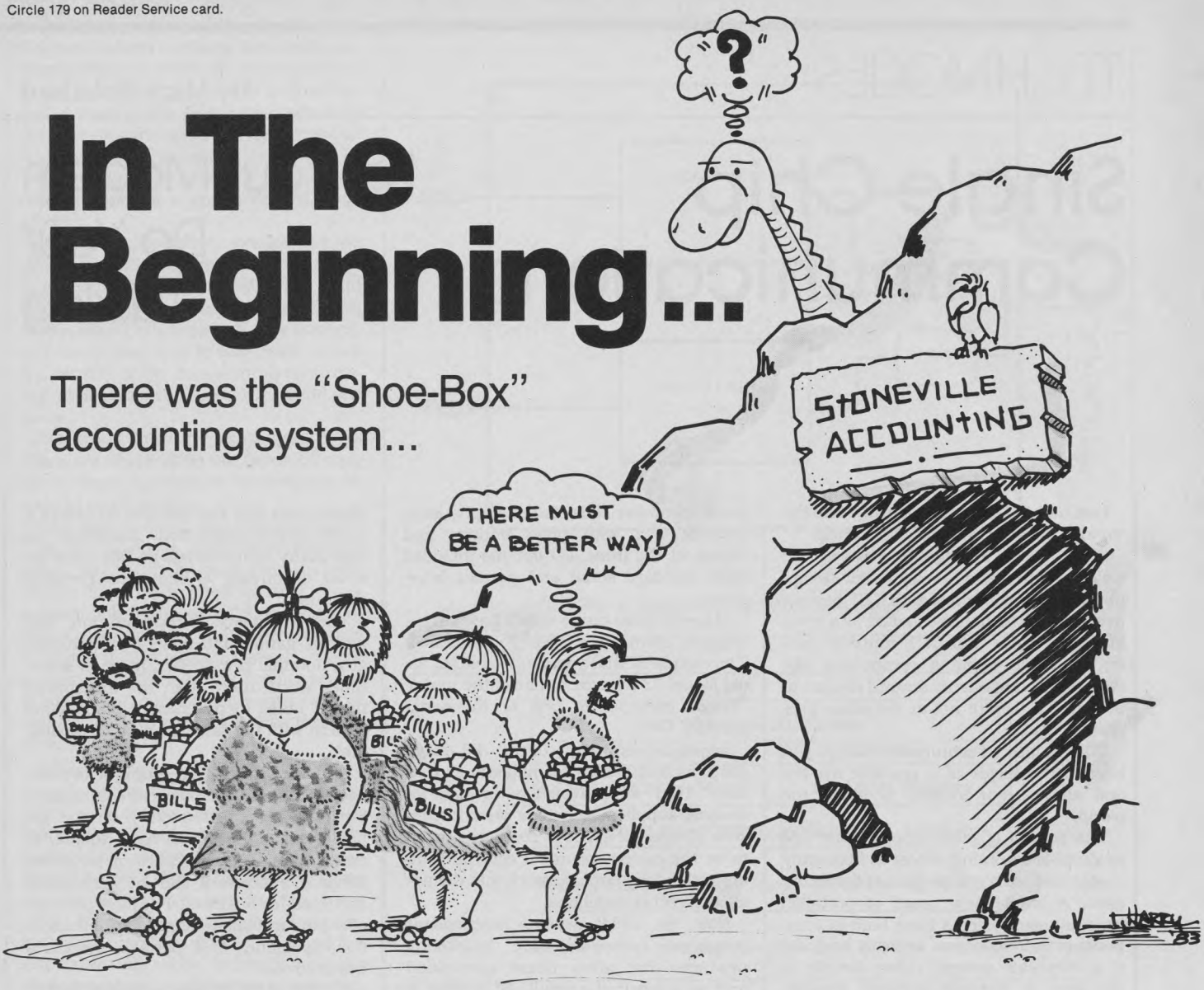
Reference Encyclopedia For The IBM Personal Computer
Gary and Karen Phillips
Ashton-Tate, 1983
Culver City, CA 90230

Set-FX
SoftStyle, Inc. ($59.95)
Suite 200
7192 Kalanianaole Highway
Honolulu, HI 96825

The Idea Processor ($295)
Idea Ware, Inc.
225 Lafayette Street
New York, NY 10012

In The Beginning...

There was the "Shoe-Box" accounting system...



But Now There's Mini-Ledger

While those dinosaur-size companies resort to large, integrated computer accounting packages, your modest-size company can track key income and expense activities through the use of Paradigm's **single-entry Mini-Ledger** program. And you can do this for only $150.00.

With **Mini-Ledger,** you can define up to 99 codes like your company car expenses, entertainment, rent, payroll, cash flow...it's up to you. And because **Mini-Ledger** is so easy to use, you don't need any accounting experience to run it. Also, several printouts are available to help you prepare clear and accurate statements of your financial operation. That's ideal for tax purposes.

To employ **Mini-Ledger,** you need either CP/M, CP/M-86 or MS-DOS, at least one disk drive, and a 132 column printer. A demo version with user's-manual is available for $30.00.

With Mini-Ledger, those prehistoric days of Shoe-Box accounting are over.

PARADIGM CONSULTANTS

MINI-LEDGER

39243 Liberty Street, Suite L • Fremont, CA 94538 • (415) 796-0543

By Mark Robillard

Single-Chip Communications

Let Your Modem Do Your Talking

Last month we began a journey into the world of portable systems engineering.

While going through that world, we looked at the various reasons for design philosophies that employed LSI components. In particular, we looked at a revolutionary single-chip CRT terminal controller. When using a component like this, other components around it seem to be bulky if they're not at the same integration level.

This month "Techniques" brings you beyond the inside of a portable system and allows your system to touch the world.

The subject of telecomputing and LSI modems is becoming almost as frequently mentioned as that of personal computing itself. A whole new breed of portable, briefcase-size systems have built-in components that allow you to easily hook up to a telephone system, either directly to the lines or through portable acoustic couplers for systems where plug-in telephones are not available.

Telecommunications

It wasn't long ago that the typical personal computer consisted of a fairly large metallic box with push buttons and lights on the front. I/O options included only standard serial and parallel interfaces.

Many of these boxes didn't have an intelligent terminal or CRT-type interface. They required that you connect either an old Model 33 teletype or one of the newer "glass" teletypes to one of the serial ports.

Personal computer users did have some limited access to modems at that time. They were available through industrial suppliers and they were expensive. The use of acoustic couplers proved to be the easiest means of communicating with a telephone network that did not allow direct connections.

Now, in 1984, several independent telephone companies are marketing products that allow direct connection and, as a result, the small LSI modem is abundant among personal computer users.

It's now possible, with a simple $400–$600 ASCII terminal, to connect to some of the larger databases through the telephone lines and emulate the actions of a much larger, stand-alone computer. In fact, many computerists go this route instead of buying a dedicated personal computer.

The availability of programming languages, games and general information at a nominal price-per-hour fee is sometimes worth more than the investment that it takes to purchase a stand-alone system with the same amount of capabilities.

Portable systems are getting a lot more use throughout the business community. Business people, frequently on the go, find that computing in hotel rooms, in restaurants and sometimes on airplanes (when it's allowed) can be convenient and good for business. The incorporation of a built-in telephone modem enhances the capabilities of a portable system a hundred-fold.

Several of today's lap portables feature LSI modem technology. A few, like the NEC PC-8201, have opted not to put in a modem; however, you'll find that the use of such portables will diminish as fully incorporated systems flourish.

Modem Fundamentals

The name "modem" is derived from the device's function: modulation and demodulation of a carrier signal.

The device itself is basically a sophisticated analog-to-digital, digital-to-analog converter. It consists of a transmitter that takes digital signal inputs and converts them into an analog carrier frequency that is modulated by another frequency. This signal is then matched to the telephone line impedance through a line transformer.

The opposite results come about through the receiver. This analog, multifrequency signal is converted into digital 1s and 0s and passed back to the com-

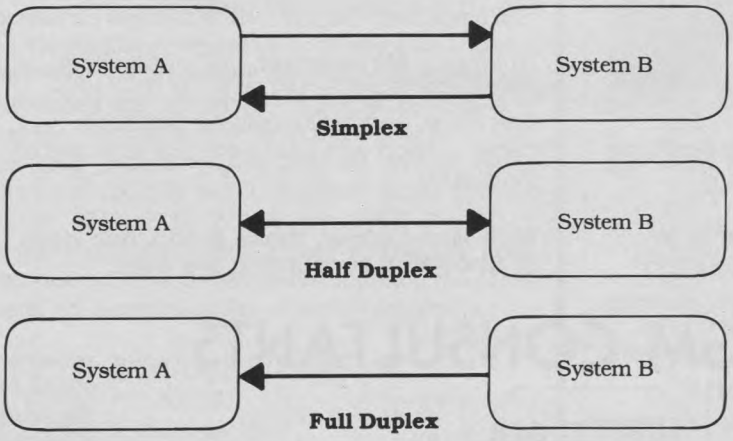


Fig. 1. Three basic computer communications modes.

Address correspondence to Mark Robillard, MJR Digital, PO Box 630, Townsend, MA 01469.

puter. Most modems utilize the RS-232C interface, which connects into most personal computer serial I/O channels. Normally you would expect this modulation and demodulation to be transparent to the user, but the nature of the telephone lines makes this a complex task.

At this time let's take a look at the characteristics of a simple Bell 103-type modem.

Modems are specified, normally, by the speed at which they operate. This speed is stated in bit rate (also called the baud rate), or bits per second. The speed must match the bit rate output of your computer's serial interface. It also must match the bit rate of the transmitting or receiving modem, depending on which end you are at.

What do I mean by which end? Well, there are three different types of communications modes that modems can be used in (see Fig. 1).

The first mode is the simplex mode. This is usually a hard-wired arrangement, but it can be used over the telephone lines. It's a one-way only transmission link from system A to system B. This type of modem would be used possibly in a data-acquisition system, where accumulated data from a remote sensor is sent in a continuous stream to a host computer. There is no interaction between the two.

When interaction must exist and only a single line is used, then the next mode, called half duplex, is often used. With half duplex, communication may be bidirectional but may exist only in one direction at a time. This is because the same circuitry is used to receive and transmit.

The final mode, which is the most common and the most flexible, is called full duplex. In this mode, bidirectional communications can exist between system A and system B. Both systems may receive and transmit simultaneously.

Most low-cost modems emulate these Bell 103-type modes. This modem lets you use all three communications modes. Normally the half duplex and full duplex modes are used; therefore there is usually a switch on the modem that allows you to select which one you want.

Modem Operation

Having gone through the various communications modes between computers, let's now look at the basic operation of a modem. Fig. 2 shows a typical modem system. Input data from a computer or terminal, represented as digital logic 1s and 0s, enters into the section called the modulator. Once in the modulator, the logic 1 or 0 is synthesized into a frequency that is then output to the line-matching transformer.

The difference between a 1 and a 0 is a shift between a high frequency and a low frequency. For this reason, the type of modulation used in a standard voice-type modem is called frequency-shift keying, or FSK.

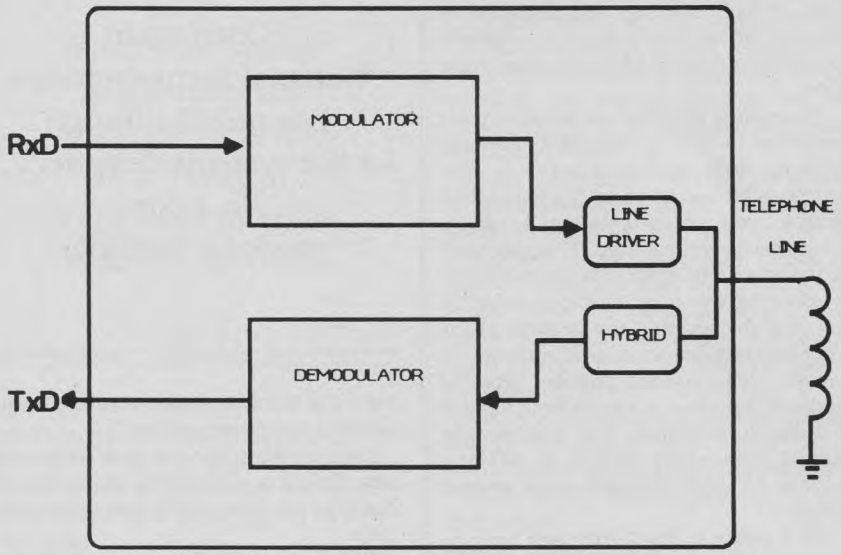


Fig. 2. Basic block diagram of a typical mode.

Bell 103 Tone Allocation		
	Originate	Answer
Data	Transmit Receive	Transmit Receive
Space	1070 Hz 2025 Hz	2025 Hz 1070 Hz
Mark	1270 Hz 2225 Hz	2225 Hz 1270 Hz

Fig. 3. Table of tone pairs used for Bell 103-type modems.

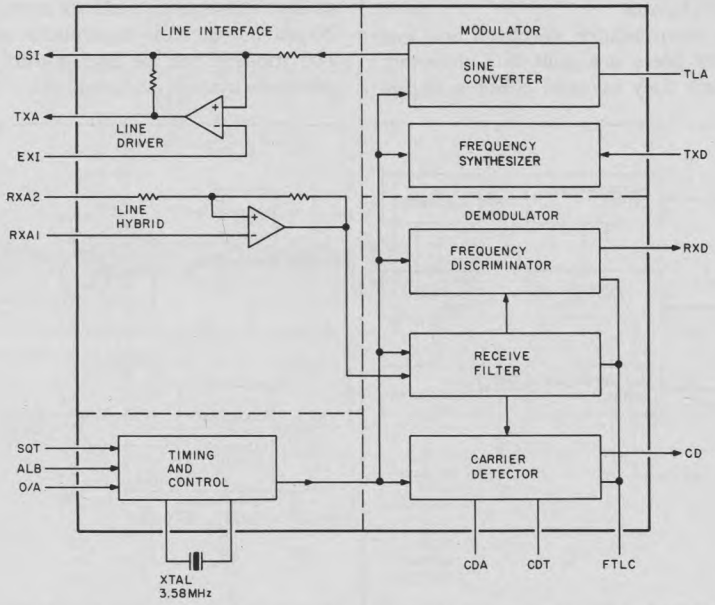


Fig. 4. Functional block diagram of the National Semiconductor MM74HC943.

There is a set of two frequencies used to receive and transmit both a 0 and a 1. This results in four different frequency tones. Which tone is used depends on whether the modem is set up to originate communications or answer communications.

The modem that you use on your home computer to talk to standard database systems, such as CompuServe or The Source, is known as an originate modem, because you are originating a call to some answering machine. This machine, on the other end, responds to your query by allowing you to enter log-on information. It's obvious that the modem at the other end is called an answer modem.

Most Bell 103-compatible personal modems allow you to switch from answer to originate and back. Fig. 3 shows the various frequencies used in a 103-type modem for both originate and answer modems.

You'll notice in Fig. 3 that data is represented by a space or a mark. In the telecommunications world, the references to mark and space have been around for some time. The digital logic equivalents of these are: a logic 1 equals a mark condition and a logic 0 equals a space condition.

It is not a trivial task to synthesize an exact frequency, such as the transmit frequency, for a mark entering an answer modem. A value of 2225 Hz is not your run-of-the-mill standard crystal frequency. It is, however, a standard frequency used in modems. For years people have synthesized these frequencies using a great deal of analog and digital components. It's now possible to obtain modem circuitry that is contained on a single integrated circuit.

These tones that are output from the modulator eventually find their way in through the receive circuitry of the modem. This receive circuitry is known as a demodulator.

In the demodulator, the high and low frequency pairs are split and detected. From here they become either a digital logic 1 or 0 (mark or space) back to the answering computer system.

Demodulation is done by a series of filters, detectors and comparators. The circuitry involved in this is also rather complex.

Within the last year or two several integrated circuit manufacturers have devised ways of putting the modulator and demodulator portions of the circuitry on a single chip. Many of these chips require external components, such as op-amps and filters, to complement the digital circuitry included on the chip.

There's a new chip, however, that incorporates all of the necessary circuitry to implement a Bell 103-type modem on a single 20-pin IC.

Once again National Semiconductor has pulled through for the systems designer . . . No, I don't work for National.

National Semiconductor MM74HC943

Once again, National Semiconductor has pulled through for the systems designer (no, I don't work for National).

Last month we looked at how National had designed a single-chip terminal microprocessor that included just about everything required to build a standard alphanumeric terminal with graphics ability. This month we'll see how, using a 20-pin IC, the full capabilities of a Bell 103 modem can be had within the lap portables system environment.

Fig. 4 is a block diagram of the 74HC943's structure. Both the modulator and the demodulator components are built into the chip. There is also a timing and control portion, which, through the use of a standard 3.579 MHz color burst crystal, provides all of the timing necessary to produce not only the frequencies but the correct mark/space ratios for transmission over the phone line.

Also included on this chip (but not on any other single-chip modem) is a line driver. This driver allows direct connection to a line-coupling transformer. It requires no external op-amps or line-driver components. Also, there's an on-chip hybrid that converts a four-wire system to a two-wire system that allows full-duplex operation at 300 bps. This modem is designed only for 300-bps use, the typical Bell 103 communication rate. It also is the most popular rate used for small computers. Databases such as CompuServe and The Source charge less money when you use a 300-bps modem instead of a faster one.

Fig. 5 provides a closer look at the modulator section. This section provides a programmable divider that synthesizes the transmit/receive frequencies. An internal clock supplies a reference signal that enters this programmable divider.

You'll notice that the modulus control has, as a second input, a pin called O/A. This pin selects whether the modem is in either the originate or answer mode; therefore, it determines which of the two frequency pairs to use.

The transmit data (TXD) and originate/answer (O/A) pins set the divisor of the dual modulus programmable divider. This provides a clock frequency 16 times the frequency of the carrier to be transmitted. The signal is then fed to a four-bit counter whose outputs go to a sine-conversion ROM.

This sine ROM acts like a 4-to-16 decoder that selects the appropriate tap on a digital-to-analog converter to synthesize an approximated sine wave. From here it goes to a low-pass filter and out to the line driver.

Inside Line Drivers

Fig. 6 shows the line driver's internals

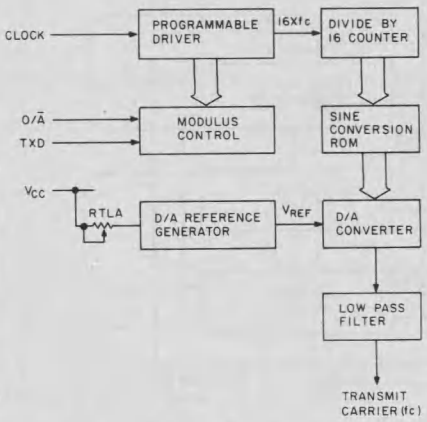


Fig. 5. Detailed block diagram of the modulator section.

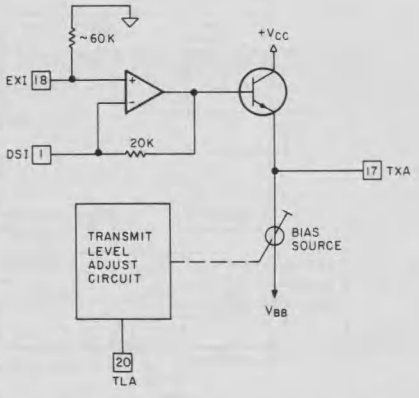


Fig. 6. Schematic representation of the line driver.

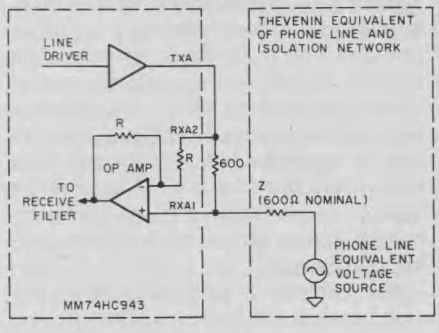


Fig. 7. Simplified schematic representation of the line hybrid.

within the 74HC943. The line driver consists of a class A power amplifier, which is used for transmitting the carrier signals from the modulator.

The schematic shows an external input that can be applied to the driver also. The output of a touch-tone generator can go into this pin, allowing you to build an autodialing-type modem.

The output of the line driver, the TXA pin, is applied directly to a line-matching transformer through a 600-Ohm resistor. This transmitted output is fed to the same point on the matching transformer that the receive input is taken from.

To discriminate between receive and transmitted outputs, a line hybrid is required. The hybrid is a circuit that performs a 2-to-4-wire conversion. It separates the transmitted output from the receive input. A more detailed view of the hybrid included on the National Semiconductor chip is shown in Fig. 7.

Under ideal conditions the telephone line and isolation network have an equivalent input impedance of 600 Ohms. Under those conditions, the gain from transmitter to the op-amp output shown in the diagram is 0, while the gain from phone line to the op-amp output is 1.

Thus, the hybrid, by subtracting the transmitted signal from the total signal on the phone line, will remove the transmitted component. Unfortunately, ideal conditions don't always exist and filtering is used to remove the remaining transmitted signal component. The hybrid, as you can see, performs the action of the split between transmit and receive inputs. An exploded view of the full demodulator section of the receiver is shown in Fig. 8.

Transmitted signals enter the receive filter; at this point the signal is coming from the line hybrid. It's a mixture of transmitted and received signals. The receive filter removes most of the transmitted signal so that only the receive signal goes into what is known as a discriminator.

This is where the tone frequencies entering the modem are turned into 1s and 0s. Fig. 9 is a functional block diagram of this discriminator; it shows that the signal going into the discriminator first goes through a limiter section, which provides a high-gain output to the remainder of the discriminator. In doing so, it also balances the energy entering the discriminator for both the mark and space signals.

From the limiter, the signal is applied to band-pass filters tuned to mark and space frequencies. The outputs of the mark and space band-pass filters are then rectified to extract the output amplitudes. Next, the signal is filtered to remove any residual ripple and low-pass filter outputs. Finally, the signal goes to comparator, which determines if the mark or space path has received greater energy and, thus, if the incoming data is a mark or a space. Demodulated digital data exits the discriminator as a TTL-level signal.

What's nice about single-chip LSI technology is the fact that you don't have to know the inside operation.

Practical Information

For those of you who are not inclined to

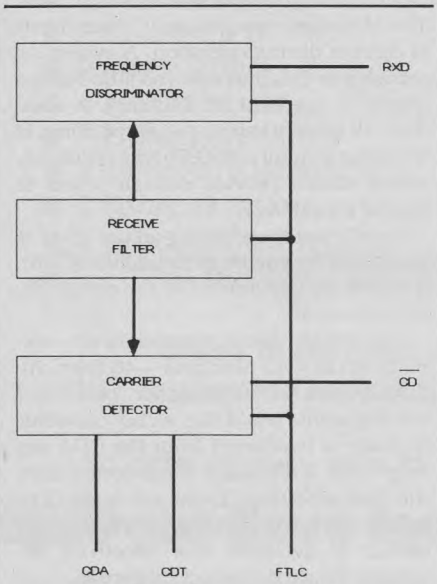


Fig. 8. Detailed block diagram of the demodulator section.

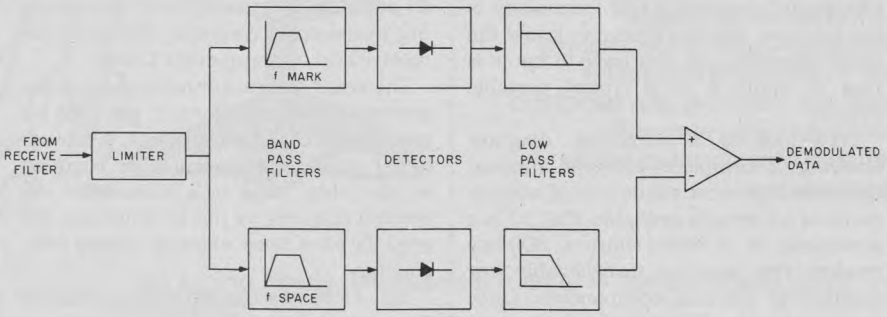


Fig. 9. Exploded view of the receive discriminator.

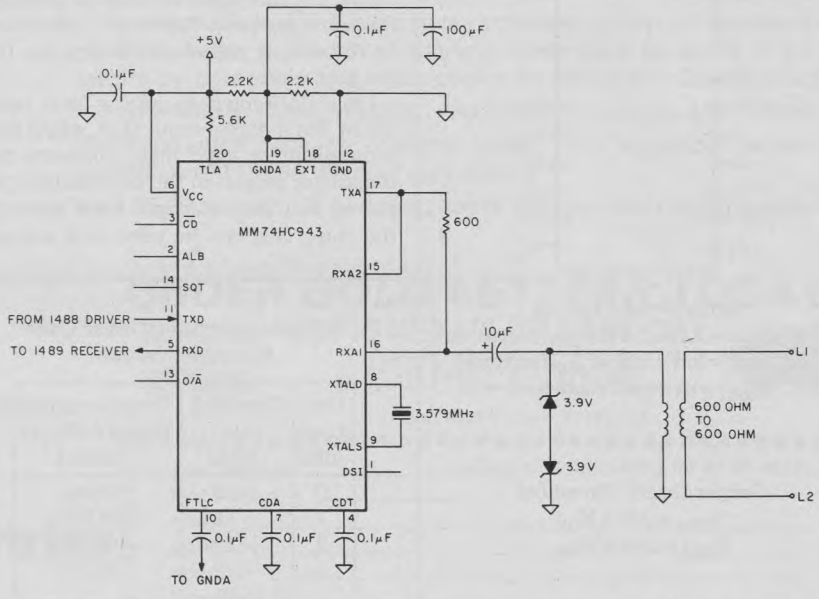


Fig. 10. Complete schematic diagram of a Bell 103-type modem using the National LSI modem chip.

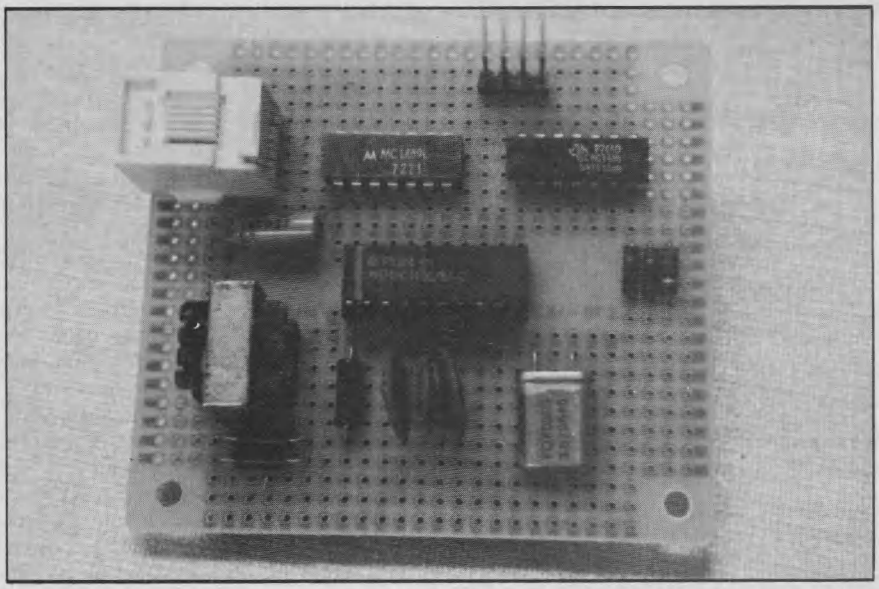


Photo of the completed project using the National Semiconductor modem chip. This is a photo of a board built to the schematic in Fig. 10.

the analog way, the last few paragraphs describing the internal operation of the modem were probably boring. What's nice about single-chip LSI technology is the fact that you don't have to know the inside operation; all you have to know is how to apply it in a typical portable system.

Let's look at a schematic diagram showing a complete 103-type modem that uses the most minimum of components of all circuits available. Fig. 10 is a schematic of a direct-connect 300-bps modem that uses an unbelievably low number of external components. Once you remove the RS-232C line driver and receivers, no external IC packages are required. The crystal pins are connected to a standard 3.579 MHz crystal, and a readily available 600-to-600-Ohm line-matching transformer connects directly to the receive and transmit output pins.

The zener diodes, connected across the primary of that transformer, are used for protection from line transients. A few external timing components are required on the chip. Now let's investigate the modem chip pin by pin to determine the need for all of these external timing components.

Fig. 11 shows the pin connections for the modem chip.

Pin 1, the driver summing input (DSI), may be used to transmit externally generated tones, such as touch-tone signals. We looked at a similar input, called EXI, on pin 18; this input can also be used for the same purpose. There are differences in the output signal, depending on the pin you select.

Pin 2 is known as analog loop back (ALB). It's a digital input that, when presented with a logic high, connects the modulator output to the demodulator input so that data is looped back through the chip. This can be used as a self-test when you're trying to determine whether your system is operating.

Pin 3 is the carrier detect output (CD). It provides a logic low level when a transmitted carrier is sensed by the carrier-detect circuit built into the chip. Many times, this output is connected to an LED, in small modems, to show when a carrier has been detected.

In modems that are acoustically coupled to the line, you may hear a tone. At this time the carrier-detect output goes low; then it's time to place the receiver into the acoustic cups. On a direct-connect system, however, it's not necessary, although it might be nice to have an illuminated indication that a carrier is detected and you are indeed on-line.

Pin 4, the carrier detect timing pin (CDT), is the first of the adjustable-timing pins we'll cover. A capacitor, which connects this pin to ground, sets the time interval that the carrier must be present before carrier detect goes low.

This allows the system to check for spurious carrier errors and allows the modem to operate only if a solid carrier signal is available. You won't want the time constant for this line to be too long because some answering systems disconnect you if they do not hear a carrier within a brief time period.

The equations relevant to the component used for the carrier-detect timing and other components are shown in Fig. 11. By going through these equations, you should be able to determine the time, in seconds, that you require, and the capacitor value (in μF) that is required to produce this time. In a typical system, a .1μF capacitor would be used; however, the final timing is up to you.

Pin 5, the receive data input (RXD), requires a standard TTL level 1 or 0 (mark or space) input.

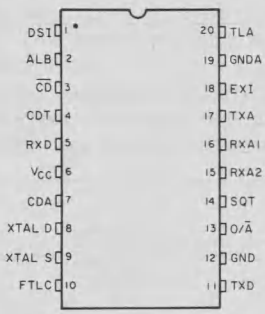
Pin 6 is the positive supply voltage pin (Vcc). A +5V supply is connected here. The chip itself requires only about 8 mA of current during operation. A system as complex as this that takes so little voltage might be operated off a battery. A standard 9V battery can be regulated down to 5V using a small LM309H-type regulator, which should provide enough power to last for a long time.

Pin 7, another timing-adjust pin, is known as the carrier detect adjust (CDA). It's used for adjustment of the carrier detect threshold.

The carrier detect trip points are normally set at −43 dBm and −46 dBm. Although you, as the designer, can select the trip points you'd like, a .1μF capacitor typically is connected from the CDA pin to ground. If a resistor is connected from the CDA pin to the CD pin, pin 3, the CDA voltage varies, depending on whether the carrier is detected; this effectively increases the carrier detect hysteresis.

As I said, though, in most applications, the .1μF capacitor is used.

Pins 8 and 9 are where the 3.579 MHz crystal is connected (XTALD, XTALS).



Carrier Detect Timing

T off to on = $3.6 \times 10^6 \times C_{EXT}$(seconds)

T on to off = $3.6 \times 10^5 \times C_{EXT}$(seconds)

(C_{EXT} = external capacitor)

Carrier Detect Threshold

$V_{CDA} = 244 \times V_{ON}$

$V_{CDA} = 345 \times V_{OFF}$

Fig. 11. Pin connection diagram of the MM74HC943 with design equations used for timing pins.

Universal Service Order Code Resistor Values

Line Loss (dB)	Transmit Level (dBm)	Programming Resistor (R_{TLA}) (Ohm)
0	−12	Open
1	−11	19,800
2	−10	9200
3	− 9	5490

Fig. 12. Table of resistance values to set transmit level output.

This generates all internal timing signals for use through the programmable divider for implementing the four sine-wave frequencies required for frequency-shift keying. No external capacitors or any other components are required to be connected to these two pins.

Pin 10, the filter test limiter capacitor input (FTLC), is connected to a high-impedance output of the receive filter. It may be used to evaluate filter performance or to evaluate the demodulator. For normal modem operation this input is ac-grounded using a .1μF capacitor. Pins 10, 7 and 4 all use .1μF capacitors in a typical application.

Let's jump ahead to pin 20.

Last Component

Pin 20 is the last analog component that must be selected for use for normal operation of the part. This pin is called the transmit level adjust (TLA); a resistor connected from TLA to +5V sets the transmit level.

Fig. 12 shows the Ohms value of the TLA resistor. You'll select one of the three resistance values or the forth value, which is an open pin, depending on the decibel level you wish to transmit at. Typically, a resistance of approximately 5.6K Ohms, translating to a −9 dBm transmit level is used (see Fig. 10).

Moving back, pin 11 is the transmitted data input (TXD). This is where the terminal or computer wishing to speak to the telephone line enters 1s and 0s.

Pin 12 is the ground line for the power supply.

Pin 13 is the originate/answer mode select (O/A). When a logic 1 is applied to this pin, it selects the originate mode of operation. This is the mode you use when you call a database. A logic low on this pin selects the answer mode, which allows your computer to answer a request from the one originating the call.

Pin 14, the squelch input for the transmitter (SQT), disables the modulator when a logic 1 is applied to it. As shown in the schematic, I have allowed you to connect this pin yourself. It could be used as a form of protection to prevent spurious transmit inputs from going out on the line. In some high-security data connections it is required to squelch the output of the transmitter during the power-down or power-up phase of a terminal.

Pins 15 and 16 are the receive analog connections (RXA2, RXA1). They are analog inputs that require connection between the transmit output, pin 17, and the line transformer. Between RXA1 (pin 16) and TXA (pin 17) is a 600-Ohm line-matching resistor, required to match the impedance of the line-output transformer.

Pin 18 (EXI), as I went over earlier, is a high-impedence input to the line driver that allows, like pin 1, externally generated tones to be fed to the line transformer.

Pin 19 is an analog ground reference pin (GNDA) that allows a reference point for the analog signals. As you can see in the schematic, this reference is set at +2.5 V, or half of the logic supply power input. This allows for a pseudo positive-to-negative swing on the output.

That does it for pin descriptions. As shown in the photograph of a circuit designed using the schematic, the circuitry requires little space; once you remove the RS-232 driver and receiver, it becomes minimal.

Another Modem Circuit

Now that I've shown you the operation and connections necessary for use with the National Semiconductor MM74HC943 modem, I'd like to take you back to show you another modem circuit that uses a similar component.

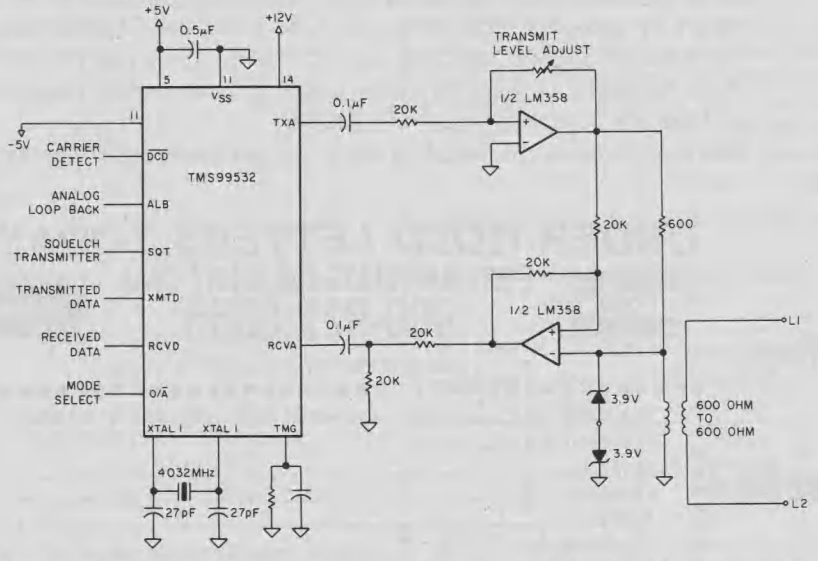


Fig. 13. Schematic diagram showing the use of the TI TMS99532 single-chip modem.

Fig. 13 is a diagram showing the use of the Texas Instruments TMS99532 single-chip modem. This chip requires external filtering, which means adding a small amount of external circuitry. Note the crystal frequency tied between the XTAL1 and XTAL2 pins. Also note the fact that this chip requires a +5, +12 and −5 voltage source. Other outputs and inputs are similar to the National Semiconductor component.

I mentioned the crystal frequency because a 3.579 MHz crystal is common in all NTSC color TVs and is probably the most inexpensive of all crystals. It's also used in low-cost personal computers; the 4.032 MHz crystal used with the TI chip, however, is neither a common value nor a cheap crystal.

The various power supplies necessary will require you to build a special power supply just for the modem. This may be a problem in a portable system that is designed for +5V use only. In a machine where memories and microprocessors use only +5V, a modem using this chip may be hard to fit in.

Close Out

Well, there you have it—a single-chip modem designed for use in portable systems. It is based on CMOS technology, runs at 300 bps and requires only a +5V supply. I'm sure, in your mind, you are designing this into every piece of equipment you have. Everyone else is; why shouldn't you?

Speaking of design-ins, next month we take the single-chip CRT component that we mentioned last month and couple it with this single-chip modem to provide a complete telecommunications terminal that emulates a standard multifunction alphanumeric terminal with some graphics ability. Also included next month will be a discussion of several letters that have arrived requesting information on some subjects.

Special thanks should go to Bill Kofoed of National Semiconductor for providing the timely information on this and other National Semiconductor components. A person who has worked directly with him to provide this information is Pat Potter of AD Systems, Inc. Many thanks to both.

Much of the design information included in this article came from the excellent documentation available on the MM74HC943. I suggest that you contact your local National Semiconductor representative and ask for all of the data sheets on this modem chip, including application note 347, which is known as the Design Guide.

Between this and the comparison report showing the 943 versus the TI part, you'll have enough information to design what you want using this part.

I look forward to joining you next month as we close out this miniseries on portable systems, showing the implementation of the ultimate LSI lap portable.□

The Latest Style

The DTC Style Writer™ should be your daisy wheel printer for professional letter-quality printing, when presentation of you and your information is important.

- 35K Buffer Memory
  frees computer from driving printer
- Automatic Proportional Spacing
  for professional appearance
- Graphic Quality
  for chart & graph applications
- Bi-directional Printing
  for increased print speed
- Cartridge Print Wheel
  for font variety, easily changed
- Toll Free 800 Telephone Number
  for readily answered questions
- Options
  Buffer Memory Expandable to 67K
  Sheet Feeder
  Forms Tractor

DTC means quality, service & support.
Drive a DTC Style Writer at your computer store, start printing with style for only $899.

DTC

Circle 310 on Reader Service card.

Data Terminals & Communications
590 Division Street, Campbell, CA 95008
800-962-8185 • In CA 800-538-9294

Distribution Inquiries Welcome

By Robert W. Baker

Make Your C-64 A Copy Cat

This Emulator Makes PET and CBM Software C-64 Material

Converting Disk Commands

The PET Emulator makes a large number of PET and CBM software available for the Commodore-64. It doesn't, however, do anything to handle the Basic 4.0 disk commands that many of these older programs may have used. Another possible problem is the use of drive 1 in the 4040, 8050 and similar disk drives. Fortunately, many of these problems can be fixed if you're willing to spend a little time converting a program.

Many utility programs used both disk drives on the older dual drive units for convenience, reading from one drive and writing to the second drive. Most of these programs don't need to use both drives and it's a simple matter to change the drive number in one Open command.

Some programs need two disk drives due to the amount of data or the size of the program with all its overlays or whatever. In this case, besides changing the drive number, you can change the device number in the same Open command to use a second VIC-1541 disk, for example, as device 9. You'll probably have to open another command and error channel for the second disk device, so be sure that error checks are made to the correct logical file.

A Basic Assumption

All of this assumes, of course, that the Basic program doesn't use Basic 4.0 disk commands. As long as the newer disk commands aren't used, everything should work correctly on the C-64. If you do encounter a program that uses the newer disk commands, don't despair; you can still convert most of the commands to a form that will work on the C-64.

The first problem area you're likely to encounter is the DOpen command. This command is easy to convert to the older and more general Open command, as long as you know the DOpen parameters. To convert any general DOpen command:

```
DOPEN#f,"filename"
DOPEN#f,"filename",Dd,Uu
DOPEN#f,"filename",Dd ON Uu
```

Use the same logical file number (f) and filename. The disk unit number (u) probably won't be specified, so use the standard default value of eight unless you're using a second 1541 drive.

The drive number (d) may or may not be indicated; if omitted it defaults to zero. Since you have to use drive #0 for any 1541 disk, you should always use zero for the drive number.

Now, substitute the correct numbers in the appropriate places to build the corresponding Open command:

```
OPEN f,8,s,"0:filename"
```

Don't forget to add the appropriate secondary address(es) following the unit number. This number should be in the range of two to 14, since zero, one and 15 have special meanings to the disk controller. Zero and one are used for Load and Save operations, while 15 is used for the special command and error channel. To avoid problems, you should try and use different secondary addresses for each file open at the same time.

There are normally a few other parameters that follow the filename in the Open command. They define the file type and how it is to be used, in that order. The DOpen command is somewhat limited–you cannot open a program or a user file with DOpen. The Open command has more capabilities, including some undocumented functions.

If the DOpen command has an Lrrr parameter, then the file is a relative file and the number (rrr) following the L defines the record length. In this case, convert:

```
DOPEN#f,"filename",Dd,Lrrr
```

to the more general form:

```
OPEN f,8,s,"0:filename,L," + CHR$(rrr)
```

When there is no Lrrr parameter in the DOpen command, as shown in some of the earlier examples, the file being opened is a sequential data file. There may be a W parameter in the DOpen command in this case. If present, the W indicates a new file is being created and will be opened for writing. To open a file to be created, replace the DOpen command:

```
DOPEN#f,"filename",Dd,W
```

with the corresponding Open command that has the added sequential file type (S) and write mode (W):

```
OPEN f,8,s,"0:filename,S,W"
```

If the W parameter is missing in the DOpen command, then an existing data file is being opened for reading. In this case, convert the DOpen command:

```
DOPEN#f,"filename",Dd
```

to the general form with the sequential file type (S) and read mode (R):

```
OPEN f,8,s,"0:filename,S,R"
```

If in any of the DOpen commands, the filename is specified by a variable:

```
DOPEN#f,(FI$),Dd
```

then you can use the same variable in the corresponding Open command:

```
OPEN f,8,s,"0:" + FI$ + ",S,R"
```

Another Way to Do It

This is true for most of the disk commands, so I won't try to bore you by repeating myself. Instead, let's look at another way the Open command can replace a Basic 4.0 command.

If a program wants to append data to the end of an existing sequential data file, you may see an append command used. The general form is:

```
APPEND#f,"filename",Dd
```

This command can be converted to an equivalent Open command by using an A for the file type following the filename:

```
OPEN f,8,s,"0:filename,A"
```

Address correspondence to Robert W. Baker, 15 Windsor Drive, Atco, NJ 08004.

The Append mode (A) implies that a sequential data file is opened for writing, but that the previous data is left unchanged.

Besides replacing the existing DOpen commands, you'll probably also have to add an Open command to be able to use the command and error channel (secondary address 15). This channel must be opened for issuing special disk commands and checking the disk error status. For convenience, most people use a logical file number of 15, the same as the secondary address. Thus, you simply need to add an:

OPEN 15,8,15

somewhere near the start of your program.

Once all the correct files are opened, all the GET#, INPUT#, and PRINT# commands that handle disk files should work without any additional changes. However, most Basic 4.0 programs will test the special disk status variable DS after disk operations to check for errors. An error message might even be displayed using the disk status string DS$.

The values of DS and DS$ are normally automatically obtained by Basic 4.0 from the disk drive. All the program has to do is reference the variables when needed. Without Basic 4.0, you might want to read the error channel after each disk operation using something like:

INPUT# 15,EN,EM$,ET,ES

Now you'll have the same information available to use as needed. If the program tests the value of DS, you can then substitute EN for DS. If DS$ is used by the program, then try replacing DS$ with EN;EM$;ET;ES to get the same information.

Time to Close

When the program is done with a file, that file must be properly closed. This is especially important with files being written to disk. You must be sure that all data is actually written on the disk. Basic 4.0 lets you use the DClose command for disk files followed by the appropriate logical file number (f). If DClose is used without any number, all currently open disk files are closed.

All DClose commands can be replaced with a Close command with the same logical file number. If no number is specified, replace the DClose with separate Close commands for each logical file used. Also, be sure that the command and error channel is closed.

More on D Commands

The DLoad and DSave commands are closely related to the DOpen command. Although these aren't normally encountered within a Basic program, it's possible when using program overlays. The general command forms:

DLOAD Dd,"filename"
DSAVE Dd,"filename"

are easily converted to the older command format:

LOAD "filename",8
SAVE "filename",8

This takes care of the Basic 4.0 disk commands that can be converted to equivalent Basic 2.0 commands. The remaining 4.0 disk commands have to be converted to commands issued via the disk command channel (secondary address 15).

These commands are issued using a PRINT# command with the appropriate logical file number for the command channel that was opened. If you use an OPEN 15,8,15 to open the command and error channel, use PRINT# 15. . . to issue commands and INPUT# 15. . . to read error status.

Commands in Record Numbers

When a program uses relative files, you'll probably see a number of Basic 4.0 Record commands. These commands are used to position the relative file pointer to the start of any specific record within the relative file, or to any byte position within a record. The general form of the command is:

RECORD#f,r,b

to position the logical file (f) to the specified record (r) and byte (b). This command can be replaced with:

PRINT 15,"P" + CHR$(s) +
CHR$(r-10) + CHR$(r-hi) + CHR$(b)

The CHR$(. . .) functions must be used to specify the various parameters since they are actually binary values passed to the disk controller. The first value following the P indicates the secondary address(es) that corresponds to the address used in the Open command of that file. The next two values indicate the 16-bit record number in 6502 address format, low byte first followed by the high byte. These values can be computed from the desired record number (r) as follows:

r-high = INT(r/256)
r-low = r – (r-hi * 256)

The last value is the optional byte position within the record. It should be in the range of one to 254. If a byte pointer is not specified in the Record command, you should use a value of one for b in the replacement command.

How to Use Your Header

The Header command is used to format or "new" a blank disk. It allows you to name the disk and specify a two-character disk identifier or ID. If a disk ID is not specified, then all files are erased from a previously formatted disk and the disk is renamed as indicated. The general form of the Header command is:

HEADER Dd,"disk-name",lid

This can easily be replaced with:

PRINT#15,"N0:disk-name,id"

As previously mentioned, the two-character ID may not be present in the Header command if the program wants to erase all the files but still keep the previous disk ID. This can be used only on a disk that was previously formatted. When an ID is not specified, the disk directory is cleared but the disk is not reformatted. In this case, change:

HEADER Dd,"disk-name"

to the older format:

PRINT#15,"N0:disk-name"

The Scratch command is used to delete one or more files from disk. The general form is:

SCRATCH Dd,"filename"

and can be changed to:

PRINT#15,"S0:filename"

The Rename command is used to change the name of any disk file. The general form of this command is:

RENAME Dd, "old-name" TO "new-name" and can be changed to:

PRINT#15,"R0:new-name = 0:old-n-

Be careful, watch the reversed order of the filenames between the two command forms.

The Copy command provides selective copying of any disk file. This command cannot be used to copy files from one disk unit to another, but it is powerful. A simple copy of any file is accomplished by:

COPY Dd,"file-1" to Dd,"file-2"

and can easily be converted to the corresponding:

PRINT#15,"C0:file-2 = 0:file-1"

Note the reversed order of the filenames between the two command forms.

In this example, both files exist on the save disk drive. The Copy command can be used to copy files from one drive to another but only on dual drive units, such as the 4040 and 8050 disks. You cannot use a Copy command to copy files from one 1541 disk to another. For instance, an attempt to copy from one drive to another:

COPY D0,"file-1" to D1,"file-2"

cannot be easily done on the C-64. Hopefully, the program can be converted to use a single drive. If two drives are needed and you've changed the drive 1 references to a second unit, a copy cannot be done by the disk controller. You might try:

1. Opening the original file for reading.
2. Opening the new file for writing.
3. Reading the original file byte-by-byte and writing to the new file as it is read.
4. Closing both files when done.

One other comment while on the Copy command. Be aware that you can also do a full disk copy on dual disk drive units by issuing:

COPY D0 to D1

Obviously, this cannot be easily converted for the C-64 with 1541 disks, but the command can be changed to:

PRINT#15,"C1=0"

Fortunately, this command wouldn't normally be used under program control.

The Concat command is used to concatenate sequential data files into one single file. It cannot be used with any file types other than sequential data files. The general form is:

CONCAT Dd,"file-1" TO Dd,"file-2"

and will add the data in file 1 to the end of the existing data in file 2. This command has an unfortunate side effect. It deletes file 1 if it resides on the same disk as file 2.

Without Basic 4.0, you'll need to replace the Concat command with a Copy command that copies and concatenates the files:

PRINT#15,"C0:file-3=0:file-1,0:file-2"

In this instance, the newly created file (file 3) cannot have the same name as the original file or you'll get a disk error (FILE EXISTS). If the original Concat command would have deleted a file, you may have to scratch a file after using the Copy command.

Actually, the Copy command is more flexible since you can concatenate up to four files at one time:

PRINT#15,"CO:file-x=0:
file-1,0:file-2,0:file-3,0:file4"

When you enter this many filenames in the Copy command, be careful. You may need to use short filenames, because the entire command string must be limited to a maximum of 40 characters!

WHAM, BAM . . .

The Collect command is used to validate a disk and recreate a valid Block Availability Map (BAM). All files are checked by reading consecutive sectors and rebuilding the BAM. Any improperly closed files are deleted and the updated BAM is rewritten on the disk when the command is done. The Basic 4.0 syntax is simply:

COLLECT Dd

and can be easily replaced with:

PRINT#15,"Vd"

to accomplish the same function. On the older disks, the Collect function was called Validate, thus the reason for the V character.

Basic 4.0 and the dual disk drives also provide a Back-up command for creating a duplicate disk for back-up purposes. This can only be done with two disk drive units and cannot be used with the VIC-1541 disks. If you have some means of connecting a 4040, 8050 or similar dual disk drive to the C-64, then you can change the Backup command from:

BACKUP Dx to Dy

to the equivalent:

PRINT#15,"Dy=x"

Watch the reversal of the disk drive numbers!

When backups aren't needed under program control, you may be able to use the Copy/all program (by Jim Butterfield) or other unit-to-unit, disk-switching copy programs.

The remaining Catalog and Directory commands are used to display the disk directory. With Basic 4.0 this could be done under program control without destroying the program in memory. With the C-64, the normal method is to load the directory and then list it. You can, however, write a routine to open the directory as a file by using a filename of $0 and then read the data. This does tend to get a little messy and may not be worth the effort.

Data Builder

Here's a handy utility for all Commodore systems. Although written for the PET and CBM systems, it will also run on the VIC-20 and C-64 (Listing 1).

The program is designed to read a machine language program from disk and create a Basic program on disk with the same machine language program converted to data statements. The program also adds a for-next loop with the correct parameters to read the data and poke it into memory. Now you have an easy way to get machine language programs into a form that can be merged with a Basic program.

When you run the program, it first asks for the filename of the machine language program to be read and converted (lines 210–260). The machine language program must be a standard, loadable program file. It cannot be any kind of intermediate file created by an assembler. If you use an assembler to create the machine language program, you can load it by following normal procedures then use a monitor to save it on disk as a program file. Once it's saved as a program file, this program can be used.

What It Does

After the program knows the machine language program's filename, it prompts for the name of the new Basic program to be created (lines 260–310). There cannot be any file on disk already using this name. If either file cannot be opened, an error message is displayed and the program terminates.

Note that the Open commands in lines 250 and 300 open a program (P) file for reading and writing, respectively. This cannot be done with the Basic 4.0 DOpen command. The standard Open command must be used as shown.

You probably won't find this in the Commodore documentation, but you can

open and use program files just like sequential data files. Just keep in mind that the first two bytes of a program file specify the load address, indicating where the program will be loaded in memory.

Once the appropriate files are opened, this utility reads and displays the load address of the machine language program (lines 330–370). When the program file is opened for reading, the first two bytes read are the load address in 6502 format (low byte/high byte). Thus, the address is converted to its decimal value by adding the first byte (the low byte) to 256 times the second byte (the high byte).

A load address of 1025 is then placed at the start of the Basic program being created with the PRINT#2 in line 390. The 1025 value is the standard load address for all Basic programs on the PET and CBM systems. If you have a VIC-20 or C-64 system, there's no need to change this value since the Basic loader will properly relocate the program when it's loaded on these systems. Thus, this value works for all Commodore systems.

Entering the Main Loop

After the utility program knows the starting address, it enters the main loop, which reads a byte from the machine language program (line 410), gets the decimal value of the byte (line 420) and adds the data to the current Basic program line being constructed in L$ (lines 430–440). The byte count for the length of the machine language program (NB) is also incremented.

The length of the Basic program line created is checked in line 450 to see if more data can be added. If there's still room for more data on the same line, the program returns to line 410 to read the next byte from the machine language program. Otherwise, the subroutine at line 580 is called to add this line to the Basic program that is being written to disk. After writing the line to disk, the program returns to line 400 instead of line 410 to place the Data token (131) at the start of the Basic line.

The subroutine in lines 580–610 adds the length of the line in L$, plus the five-byte overhead for every Basic line to a pointer in LK to compute the link or starting address of the next Basic program line. The two-byte link value is written to the Basic program file followed by the two-byte Basic line number from LN. The actual line from L$ then follows, along with a zero byte end flag to indicate the end of the Basic program line. Another subroutine in lines 620–630 is used to convert the link address and Basic line number into two-byte 6502 address format and write them to the Basic program file.

Just One Last Byte

When the utility program reads the last byte of the machine language program file and detects the end of file, any remaining data will be output to the Basic program file (lines 470–480). The status from ST is saved in SS after every read to the machine language file. This value will be 64 when the end of the file is reached.

After all data has been written to the Basic program file, a Basic for-next loop will be created in L$, inserting the length of the machine language program as the loop count and the load address read used as the poke offset address (lines 500–540).

This program line is then written to the Basic program file along with a zero link (two bytes, both zero) to indicate the end of the Basic program created. Before closing all files, the utility program indicates the length of the Basic program created.

I've included a short example of the type of program the utility program creates. Note that the line numbers of the Basic program created start at 10 and increment by ten. If you want to change the starting line number, simply redefine the value of LN in line 180 as desired. Likewise, changing the value of LI in line 190 will change the increment between line numbers.

As I mentioned, the program creates Basic program lines limited to 78 or fewer characters when displayed on the screen. This allows you to use the screen editor to change the lines if necessary. If you want things compact and don't care about not being able to edit lines, you can change the test value in line 450 from 65 to 250. The larger value will cause the utility program to create the maximum length program lines that Basic can handle. The lines can be displayed but cannot be edited!□

```
100 REM ********************************
110 REM
120 REM     D A T A   B U I L D E R
130 REM
140 REM          BY: ROBERT BAKER
150 REM
160 REM ********************************
170 :
180 LN=10: REM STARTING BASIC LINE NUMBER
190 LI=10: REM LINE NUMBER INCREMENT
200 :
210 PRINT"□MACHINE LANGUAGE PROGRAM"
220 PRINT"TO BE CONVERTED IS -": PRINT
230 INPUT F$
240 OPEN 15,8,15
250 OPEN 1,8,5,"0:"+LEFT$(F$,16)+",P,R)"
260 INPUT#15,EN,EM$: IF EN<>0 THEN PRINT"DISK ERROR -";EN;EM$: GOTO 650
270 PRINT: PRINT"BASIC PROGRAM TO BE GENERATED"
280 PRINT"SHOULD BE CALLED -": PRINT
290 INPUT F$
300 OPEN 2,8,6,"0:"+LEFT$(F$,16)+",P,W"
310 INPUT#15,EN,EM$: IF EN<>0 THEN PRINT"DISK ERROR -";EN;EM$: GOTO 650
320 PRINT: PRINT"OK, CREATING NEW PROGRAM FILE...": PRINT
330 GET#1,C$: IF ST<>0 THEN 640
340 AD=0: IF C$<>"" THEN AD=ASC(C$)
350 GET#1,C$: IF ST<>0 THEN 640
360 C=0: IF C$<>"" THEN C=ASC(C$)
370 AD=AD+(256*C): PRINT"STARTING ADDRESS =";AD: PRINT
380 LK=1025: NB=0
390 PRINT#2,CHR$(1);CHR$(4);
400 L$=CHR$(131): REM "DATA" TOKEN
410 GET#1,C$: SS=ST: IF SS<>0 THEN 470
420 C=0: IF C$<>"" THEN C=ASC(C$)
430 IF LEN(L$)>1 THEN L$=L$+","
440 L$=L$+MID$(STR$(C),2): NB=NB+1
450 IF LEN(L$)<65 THEN 410
460 GOSUB 580: GOTO 400
470 IF SS<>64 THEN 640
480 IF LEN(L$)>1 THEN GOSUB 580
490 IF NB=0 THEN 650
500 REM FOLLOWING LINES CREATE A BASIC LINE
510 REM FOR X=0 TO ...:READ C: POKE...+X,C:NEXT
520 L$=CHR$(129)+"X"+CHR$(178)+"0"+CHR$(164)
530 L$=L$+MID$(STR$(NB-1),2)+":"+CHR$(135)+"C:"
540 L$=L$+CHR$(151)+MID$(STR$(AD),2)+CHR$(170)+"X,C:"+CHR$(130)
550 GOSUB 580: PRINT#2,CHR$(0);CHR$(0);
560 PRINT"LENGTH =";NB+1;"BYTES": PRINT
570 PRINT"DONE CONVERSION": GOTO 650
580 L=LEN(L$): LK=LK+5+L: X=LK: GOSUB 620
590 X=LN: GOSUB 620: LN=LN+LI
600 FOR X=1 TO L: PRINT#2,MID$(L$,X,1);: NEXT
610 PRINT#2,CHR$(0);: RETURN
620 X1=INT(X/256): X2=X-(X1*256)
630 PRINT#2,CHR$(X2);CHR$(X1);: RETURN
640 PRINT: PRINT"DISK ERROR, PROGRAM ABORTED"
650 CLOSE 1: CLOSE 2: CLOSE 15
READY.


 10 DATA69,78,32,50,69,82,42,255,0,255,0,36,48,58,68,65,84,65,32,71,69,78
 20 DATA32
 30 FORX=0TO22:READC:POKE840+X,C:NEXT
READY.
```

Listing 1. The Data Builder utility program.

LETTERS TO THE EDITOR

It's All Mixed Up

I enjoyed Greg Rogers' article "Speak Easy and Carry a Big Digitizer," (January *Microcomputing*, p. 100); however, the listings appear to be mixed up.

Herbert Vandermark
Leroy, NY

Reply:

Herbert, sorry about the confusion. We apologize for the inconvenience. We have printed the listings in their entirety (see Listing 1 and Listing 2).

Editors

What's in a Name?

In your Book Reviews section, the title of the book is always followed by the name of the author and then the publisher. However, in your Software Reviews and New Software sections, the name of the author is routinely omitted.

As a consumer of both books and software, I consider the name of the author in either medium an important identifying factor and a valuable clue about the style and quality of the product. Why does *Microcomputing* treat computer-based materials differently than printed materials in this respect?

Reply:

Your question may be better directed to software vendors. They seldom reveal who wrote a particular program. We agree that software authors should be identified.

Editors

We Noticed

We entered the program from "Playing Games with Apple Pascal" (January *Microcomputing*, p. 74). As far as we can tell, a few pages were transposed between the original program and the way it ended up in print.

We have enclosed a listing of the program with our corrections (Listing 3); as best as we can determine, the program now functions correctly. Also, we did a few modifications to the text output on the screen.

Thank you.

Don Brown
Robert Vester
Libby, MT

Reply:

Thanks for noticing our error and for taking the time to modify the program.

Editors

(The program is listed on p. 120)

Listing 1. Speak-Easy voice digitizer creates speak data modules.

```
00190           ORG     07000H                  ;ORIGINATE AT 7000H
00200 CMD       EQU     4174H                   ;JUMP VECTOR
00210 VDLINE    EQU     021BH                   ;VDLINE LINE ROM ROUTINE
00220 ;
00230 ;
00240 ;                 INITIALIZATION
00250 ;
00260 ;
00270 START     LD      HL,(CMD)                ;CURRENT VALUE IN VECTOR
00280           LD      (OLD),HL                ;SAVE IT
00290           LD      HL,CHECK                ;NEW JUMP VECTOR
00300           LD      (CMD),HL                ;STORE IT
00310           LD      A,201                   ;CODE FOR 'RETURN'
00320           LD      (START),A               ;STORE IT
00330           LD      HL,MSG1                 ;FIRST MESSAGE
00340           JP      VDLINE                  ;PUT IT ON
00350 MSG1      DEFM    'SPEAK-EASY IS NOW INSTALLED'
00360           DEFB    13
00370 ;
00380 ;
00390 ;                 CHECK FOR A SPEAK-EASY COMMAND
00400 ;
00410 ;
00420 CHECK     LD      A,(HL)                  ;GET CHARACTER
00430           CP      0D0H                    ;IS IT A DECODED '/'?
00440           DEFB    0C2H                    ;RETURN IF NOT
00450 OLD       DEFW    0                       ;WHERE TO GO
00460           RST     10H                     ;NEXT CHARACTER
00470           LD      A,(HL)                  ;GET CHARACTER
00480           CP      'T'                     ;IS IT A 'T'?
00490           JR      Z,TALK                  ;MAKE NOISE IF SO
00500           CP      'S'                     ;IS IT A 'S'?
00510           JR      Z,SPEED                 ;CHANGE SPEED IF SO
00520           RET                             ;RETURN
00530 DELA      DEFB    10                      ;SPEED
00540 ;
00550 ;
00560 ;                 TALK ROUTINE
00570 ;
00580 ;
00590 TALK      LD      HL,8000H                ;START OF TALK DATA
00600           LD      BC,6000H                ;NUMBER OF BYTES
00610 LOOP      LD      A,(HL)                  ;GET BYTE
00620           OUT     (255),A                 ;SEND IT TO SPEAKER
00630           PUSH    HL                      ;SAVE POINTER
00640           PUSH    BC                      ;SAVE COUNTER
00650           CALL    DELAY                   ;WAIT A BIT
00660           POP     BC                      ;RESTORE COUNTER
00670           POP     HL                      ;RESTORE POINTER
00680           DEC     BC                      ;DECREMENT COUNTER
00690           INC     HL                      ;INCREMENT POINTER
00700           LD      A B                     ;IS THE COUNTER TO ZERO
00710           OR      C
00720           JR      NZ,LOOP                 ;LOOP IF NOT
00730 ;
00740 ;                 START OF MAIN PROGRAM
00750 ;
00760 ;
00770 INPUT     LD      A,(3840H)               ;READ KEYBOARD
00780           OR      A                       ;IS IT ZERO?
00790           JR      Z,INPUT                 ;LOOP IF SO
00800           LD      BC,6000H                ;NUMBER OF BYTES
00810           LD      HL,BUFFER               ;START OF BUFFER
00820 LOOP1     IN      A,(0FFH)                ;GET A BYTE FROM CASSETTE
00830                                           ;MICROPHONE
00840           LD      (HL),A                  ;STORE IT IN BUFFER
00850           PUSH    BC                      ;SAVE BYTE COUNTER
00860           PUSH    HL                      ;SAVE BUFFER POSITION
00870           CALL    DELAY                   ;WAIT A WHILE
00880           POP     HL                      ;RESTORE BUFFER POSITION
00890           POP     BC                      ;RESTORE BYTE COUNT
00900           INC     HL                      ;INC. BUFFER POSITION
00910           DEC     BC                      ;DECREMENT BYTE COUNT
00920           LD      A,B                     ;TEST FOR ZERO
00930           OR      C
00940           JR      NZ,LOOP1                ;LOOP IF NOT
00950           LD      HL,MSG3                 ; 'BUFFER FULL' MESSAGE
00960           LD      DE,15360+128            ;WHERE TO PUT IT
00970           CALL    VDPUT                   ;PUT IT THERE
00980 ;
00990 ;
01000 ;                 REPLAY VOICE
01010 ;
01020 ;
01030 OUTPUT    LD      A,(3840H)               ;READ FROM KEYBOARD
01040           OR      A                       ;IS IT ZERO?
01050           JR      Z,OUTPUT                ;LOOP IF SO
01060           LD      BC,6000H                ;NUMBER OF BYTES
01070           LD      HL,BUFFER               ;START OF BUFFER
01080 LOOP3     LD      A,(HL)                  ;GET TALK BYTE
01090           OUT     (255),A                 ;OUT TO SPEAKER
01100           PUSH    HL                      ;SAVE BUFFER POSITION
01110           PUSH    BC                      ;SAVE BYTE COUNT
01120           CALL    DELAY                   ;WAIT A BIT
01130           POP     BC                      ;RESTORE BYTE COUNT
01140           POP     HL                      ;RESTORE BUFFER POSITION
01150           INC     HL                      ;INC. BUFFER POSITION
01160           DEC     BC                      ;DECREMENT BYTE COUNT
01170           LD      A,B                     ;IS IT ZERO?
01180           OR      C
01190           JR      NZ,LOOP3                ;LOOP IF NOT
01200           LD      HL,MSG4                 ;DUMP TO DISK?
01210           LD      DE,15360+192            ;WHERE TO PUT QUESTION
01220           CALL    VDPUT                   ;DO IT
01230 LOOP      LD      A,(3840H)               ;READ FROM KEYBOARD
01240           OR      A                       ;IS IT ZERO?
01250           JR      Z,LOOP                  ;LOOP IF SO
01260           BIT     0,A                     ;IS IT THE ENTER KEY?
01270           JR      NZ,DUMP                 ;GO IF SO
01280           BIT     1,A                     ;IS IT THE CLEAR KEY?
01290           RET     NZ                      ;EXIT IF SO
01300           JR      START                   ;DO IT ALL AGAIN
01310 ;
01320 ;
01330 ;                 SAVE TALK DATA
01340 ;
01350 ;
01360 DUMP      LD      HL,FILE                 ;HL POINTS TO FILESPEC
01370                                           ;BUFFER
```

Listing 1 continued.

```
01380           LD      DE,FILE+1               ;FILESPEC BUFFER + 1
01390           LD      BC,7                    ;NUMBER OF BYTES
01400           LD      (HL),32                 ;CLEAR ONE BYTE
01410           LDIR                            ;LOOP UNTIL DONE
01420           LD      HL,15360+256            ;CURSON POSITION
01430           LD      (4020H),HL              ;SAVE IT
01440           LD      HL,FILE1                ; 'FILENAME?' MESSAGE
01450           CALL    021BH                   ;PUT IT ON SCREEN
01460           LD      B,8                     ;LENGTH
01470           LD      HL,FILE                 ;WHERE TO PUT IT
01480           CALL    40H                     ;GET FILESPEC
01490           LD      HL,FILE                 ;HL -> FILESPEC
01500 FILELP    LD      A,(HL)                  ;GET A CHAR
01510           CP      13                      ;TERMINATOR?
01520           INC     HL                      ;NEXT CHARACTER
01530           JR      NZ,FILELP               ;JUMP IF NOT
01540           DEC     HL                      ;ADJUST HL
01550           LD      (HL),32                 ;TURN IT TO A SPACE
01560           LD      HL,DUMP1                ; 'DUMP' COMMAND
01570           CALL    429CH                   ;EXECUTE TRSDOS COMMAND
01580                                           ;AND RETURN TO CALLER
01590           LD      HL,MSG5                 ; 'ANOTHER' QUESTION
01600           CALL    021BH                   ;PUT IT ON SCREEN
01610 INPU      LD      B,1                     ;NUMBER OF CHARACTERS
01620           LD      HL,BUFF                 ;WHERE TO PUT IT
01630           CALL    40H                     ;GET CHARACTERS
01640           LD      HL,BUFF                 ;POINT TO CHARACTERS
01650           LD      A,(HL)                  ;GET CHARACTER
01660           CP      'Y'                     ;IS IT A 'Y'?
01670           JP      Z,START                 ;DO IT AGAIN IF SO
01680           CP      'N'                     ;IS IT A 'N'?
01690           JR      NZ,INPU                 ;GET ANOTHER IF NOT
01700           RET                             ;EXIT
01710           END     START                   ;END
```

```
00180 FILE1     DEFM    'FILENAME?                      ;FILENAME PROMPT
00190           DEFB    3
00200 DUMP1     DEFM    'DUMP                           ; 'DUMP' COMMAND
00210 FILE      DEFM    '
00220           DEFM    ' (START=8000,END=0E00H)'
00230           DEFB    13
00240 ;
00250 ;
00260 ;                 DELAY ROUTINE
00270 ;
00280 ;
00290 DELAY     LD      B,10                    ;LOOP 10 TIMES
00300 DELP      DJNZ    DELP                    ;DO IT
00310           RET                             ;RETURN TO CALLER
00320 VDPUT     LD      A,(HL)                  ;GET CHARACTER
00330           OR      A                       ;IS IT ZERO?
00340           RET     Z                       ;RETURN IF SO
00350           LD      (DE),A                  ;PUT ON SCREEN
00360           INC     HL                      ;NEXT CHARACTER
00370           INC     DE                      ;NEXT SCREEN POSITION
00380           JR      VDPUT                   ;LOOP UNTIL DONE
00390 BUFFER    EQU     8000H                   ;START OF SPEAK BUFFER
00400 ;
00410 ;
00420 ;                 IMPORTANT MESSAGES
00430 ;
00440 ;
00450 MSG1      DEFM    'Speech - by Greg Rogers'
00460           NOP
00470 MSG2      DEFM    'Press any key and SPEAK'
00480           NOP
00490 MSG3      DEFM    'Buffer is full.  Press any key to replay.'
00500           NOP
00510 MSG4      DEFM    'Press ENTER to dump to disk, any other key to try again.'
00520           NOP
00530 MSG5      DEFM    'Another Phrase (Y OR N)? '
00540           DEFB    3
00550 ;
00560 ;
00570 ;                 SET-UP CODE
00580 ;
00590 ;
00600 START     CALL    01C9H                   ;CLEAR SCREEN
00610           LD      HL,BUFFER               ;POINT TO BUFFER START
00620           LD      DE,BUFFER+1             ;POINT TO BUFFER+1
00630           LD      BC,5FFFH                ;NUMBER OF BYTES TO CLEAR
00640           LD      (HL),0                  ;CLEAR A BYTE
00650           LDIR                            ;LOOP UNTIL DONE
00660           LD      HL,MSG1                 ;FIRST MESSAGE
00670           LD      DE,15360                ;WHERE TO PUT IT
00680           CALL    VDPUT                   ;PUT IT THERE
00690           LD      HL,MSG2                 ;SECOND MESSAGE
00700           LD      DE,15360+64             ;WHERE TO PUT IT
00710           CALL    VDPUT                   ;PUT IT THERE
00720 ;
00730           RET                             ;RETURN
00740 ;
00750 ;
00760 ;                 DELAY FOR A SPECIFIED AMOUNT
00770 ;
00780 ;
00790 DELAY     LD      A,(DELA)                ;GET DELAY
00800           LD      B,A                     ;PUT IT IN B
00810 DELP      DJNZ    DELP                    ;LOOP UNTIL DONE
00820           RET                             ;RETURN TO CALLER
00830 ;
00840 ;
00850 ;                 CHANGE OUTPUT SPEED
00860 ;
00870 ;
00880 SPEED     RST     10H                     ;MOVE TO NEXT CHARACTER
00890           LD      A,(HL)                  ;GET CHARACTER
00900           CP      '('                     ;IS IT A '('?
00910           RET     NZ                      ;RETURN TO BASIC IF NOT
00920           RST     10H                     ;MOVE TO NEXT CHARACTER
00930           CALL    281CH                   ;CONVERT TO AN 8-BIT
00940                                           ;VALUE
00950           LD      (DELA),A                ;STORE VALUE IN DELAY
00960           RST     10H                     ;MOVE TO NEXT CHARACTER
00970           LD      A,(HL)                  ;GET CHARACTER
00980           CP      ')'                     ;IS IT A ')'?
00990           RET     NZ                      ;RETURN TO BASIC IF NOT
01000           RST     10H                     ;MOVE TO NEXT CHARACTER
01010           RET                             ;RETURN TO BASIC
01020           END                             ;END
```

Listing 2. Program allows you to use speak modules in Basic programs.

Who's in the WordStar Patch With Epson?

Graftrax Plus, that's who. After equipping his Epson MX-80 printer with Graftrax Plus, the author discovered that WordStar wasn't written with any of its features in mind. So he decided to "patch" up the differences.

By Tim Norton

I have just finished upgrading my Epson MX-80 by installing Graftrax Plus and have come to the harsh realization that WordStar wasn't written with any of those printer features in mind.

The backspace and overstrike work fine—so does the compressed mode for the alternate pitch. However, WordStar doesn't know that a printer can do its own underlining or double strike, so there is no way to have ↑PS (underscore) or ↑PD (double strike) just send the command to the printer. I wanted a way to use the new printer features with WordStar, and this technique lets me use almost all of them.

PRINTER PATCH AREAS		DESCRIPTION
PSINIT:	0A,0D,	Initialize printer at start of doc.
	1B,40,	<CR>, ESC @ (printer reset),
	1B,39,	ESC 9 (enable paper out sensor),
	1B,43,00,0B,00	ESC C 00H 0BH (form length = 11")
PALT:	01,0F,00,00,00	^O - Compressed mode on
PSTD:	01,12,00,00,00	^R - Compressed mode off
RIBBON:	02,1B,34,00,00	ESC 4 - Italic character set on
RIBOFF:	02,1B,35,00,00	ESC 5 - Italic character set off
USR1:	01,0E,00,00,00	^N - double size print until <CR>
USR2:	01,1B,00,00,00	ESC - escape to start command
USR3:	01,00,00,00,00	B'00' switch for some commands
USR4:	01,01,00,00,00	B'01' switch for some commands

Table 1. WordStar patches.

The Technique

When I tried to use the printer, I quickly found that the number of printer features was greater than the number of printer commands in WordStar. Most of the MX-80 options take two commands: one to enter the mode and one to return to normal. WordStar provides four user printer controls: ↑PQ, ↑PW, ↑PE and ↑PR, as well as alternate/normal pitch (↑PA and ↑PN), subscript (↑PV), superscript (↑PT), and other ribbon color (↑PY).

Each of these commands sends the value in a WordStar patch area to the printer. Values can be changed with the WordStar install program. Other ribbon color (↑PY) uses RIBBON: and RIBOFF:, which allows one printer command to be sent when changing to the new color and a separate printer command to be sent when changing back.

However, both subscript (↑PV) and superscript (↑PT) use ROLLUP: and ROLDOW:. The command sent to the printer when entering subscript mode (defined by ROLDOW:) is the same one that's sent when leaving superscript mode. You can set ROLDOW: to be the printer command that enters compressed mode and ROLLUP: to end compressed mode. However, this means that marking something in a document as

Address correspondence to Tim Norton, 1107 Southwestern Drive, Richardson, TX 75081.

being superscript would have no effect and that everything after it would be compressed.

Why not use the MX-80 superscript and subscript commands? Because this takes three commands: one to enter subscript mode, one to enter superscript mode and one to get back to normal from either mode.

Take a look at the printer commands—what features do you want? I knew I wanted italics, compressed, double-wide and emphasized print, but that takes eight commands: one to enter each mode and another to leave that mode. Those four features use up all of the WordStar patch areas and there are still a lot of printer features left.

Several programs are available to modify WordStar and give you more commands. They seem to solve the problem very well, but they cost money and I was looking for a less expensive solution.

I noticed something about the new printer commands available with the Graphtrax option. Most of them are an escape character followed by a printable character (such as a letter or number), sometimes followed by a hex one or zero. Now, if you set up the user areas so that one sends an escape, one sends a hex one and one sends a hex zero, you can build almost any command you want.

For example, if USR2:(↑PW) is an escape, USR3:(↑PR) is a hex one and USR4:(↑PE) is a hex zero, then ↑PW-↑PR will turn on continuous underlining and ↑PW-↑PE will turn it off. (See Table 1 for a description of the patches and Table 2 for a list of the printer commands.)

One Problem

The only problem you have to worry about when using this technique is that the character after the ↑PW (the "-" in the above example) does not print because it is part of a command. However, WordStar doesn't know this and will consider it printable when doing things like reformatting (↑B). It's best either to add these commands after the document is finished and formatted or to put them on a blank or short line.

Table 1 shows the patches for the user areas USR1:-USR4: as well as some of the other areas. The ESC @ in PSINIT: is a simple but useful feature that resets the printer at the start of printing each document. Now I don't have to put commands at the end of a document to cancel all of the commands used in that document. The printer is always in a known state when printing starts.

The technique of building printer commands in WordStar does have its disadvantages—sometimes surprises.

It may not be the easiest to use, but it does give me a great deal more control over the printer without having to write a formatter program. WordStar is very flexible. With these patches and a little trial and error, I can get my documents printed just the way I want them.■

WordStar command	Action
^PA	Switch to alternate character set (Compressed mode)
^PN	Switch to standard character set (Normal)
^PY	Toggles ribbon color (Italic)
^PQ	Start double wide mode, until end of line (<CR>)
^PW-^R	Turns on continuous underlining
^PW-^E	Turns off continuous underlining
^PW0	Sets line spacing to 1/8" (8 LPI)
^PW1	Sets line spacing to 7/72" (10 LPI)
^PW2	Sets line spacing to 1/6" (6 LPI)
^PW4	Turns on Italics character set
^PW5	Turns off Italics character set
^PW8	Ignores "paper out" sensor
^PW9	Enables "paper out" sensor
^PW<	Prints current line only from left to right
^PW@	Resets to power up state, including TOF
^PW#	Accepts eighth bit "as is"
^PW=	Clears eighth bit (sets it to zero)
^PWE	Turns on emphasized mode
^PWF	Turns off emphasized mode
^PWG	Turns on double strike mode
^PWH	Turns off double strike mode, super- & subscripts
^PWS^R	Turns on subscript mode
^PWS^E	Turns on superscript mode
^PWT	Resets sub-/superscript (not double/unidirectional print)
^PWW^R	Turns on double width printing
^PWW^E	Turns off double width printing

Table 2. Printer commands using WordStar patches.

Teach Your Printer To Speak Pascal

You, your Epson MX-80 and your Apple were getting along great in Applesoft Basic, but then Apple Pascal came into your life and things got tough. The author explains how to cope with the situation and teaches you to control your printer with Pascal.

By Fred Johnson

Just when I thought I was getting to know how to control the Epson MX-80 printer from Applesoft Basic, I had to go and buy the Apple Language System. It gave me UCSD Pascal, but also a lot of new problems to solve.

One of the most persistent problems I had with Pascal was the control of my printer. It didn't take me too long to learn how to transfer text files from the disk to the printer using the filer program, and transferring a disk's directory to the printer didn't take much longer, but actually *controlling* the printer from within a program eluded me for several months.

Not a Word

I read and reread the Apple Pascal reference manual, but I couldn't find one word about how to output to the printer. I found Unitread and Unitwrite. They were described as "the low-level procedures that do device-oriented I/O." This sounded like just what I needed.

I made several attempts using Unitwrite to write a procedure similar to Write or WRITELN to output to the printer. I created Print and PRINTLN procedures to output strings to the printer. This worked, but it was a real mess.

Everything had to be converted to strings first. UCSD Pascal doesn't allow me to manipulate strings quite as easily as Applesoft Basic does, but it can be done with the right procedures and functions.

The STR procedure could change an integer to a string, the CONCAT function could stick the strings together and Copy would chop the result so that it would fit the space I wanted it to. Like I said, it worked, but it was messy!

After another few weeks, I put together two pieces of information from Apple's documentation to come up with a better method of printer control.

While running the filer program, I noticed that volumes on-line shows Console:, Systerm:, Printer:, Remin:, Remout: and the disk volume names. My first clue! Somewhere in its flat little eight-bit silicon mind, the Apple knows that the printer and the disk are both I/O devices. The second piece of information came from one of Apple's sample Pascal programs called DISK I/O.

A file must be opened before Pascal can write data to the disk. Needing an open file for disk I/O is really not new to most Basic language programmers. Apple's DOS required that a file buffer be opened before any disk I/O could take place, but there were no such requirements for outputting data to the printer. This type of inconsistency in Basic meant that I had some unlearning to do.

Address correspondence to Fred Johnson, 307 Walnut St., Knoxville, IL 61448.

Consistency

UCSD Pascal, however, is terribly consistent. Since it's necessary to have a file buffer open to do disk I/O, then it must be necessary to have a file open to be able to output ASCII characters to the printer. With this idea in mind, it took me all of a couple of hours to be able to use Write and WRITELN to control the printer.

Just three types of Pascal statements were needed to control the printer. First, a file of the type text must be declared. I used P as my printer output file. This file is declared at the beginning of the program along with the rest of the variables as follows:

```
VAR P : TEXT;
```

Second, the file must be opened so that it can be written to. Opening this file is done with the Rewrite statement. The syntax of Rewrite is:

```
REWRITE( filename , title );.
```

"Filename" is the name of the output file (P in this case) and "title" tells where the data is to be transferred. As described in Apple's documentation, this would be the disk volume and file names to be used in I/O operations.

In this case, the title is 'PRINTER:'. Notice the quotes. It doesn't work if it is not in quotes. So the statement becomes:

```
REWRITE( P : 'PRINTER:' );
```

The third type of statement used to

write data to the printer is the Write or WRITELN used with the addition of a file identifier. Assuming that we are using a file named P, the Pascal statement to print the value of the variable ALPHA will look like this:

```
WRITELN( P , ALPHA );
```

Putting this all together into a small program that will print the integers from 1-10 on the printer, we have the following:

```
PROGRAM ONETOTEN;
VAR ST : STRING;
I : INTEGER;
P : TEXT;
BEGIN
REWRITE( P , 'PRINTER:' );
ST := 'THE NUMBERS ONE - TEN';
WRITELN( P , ST );
WRITELN( P );
FOR I := 1 TO 10 DO WRITE( P , I:3 );
WRITELN( P );
END.
```

The output will look like this...

```
THE NUMBERS ONE - TEN
  1  2  3  4  5  6  7  8  9 10
```

The file is opened by Rewrite, and the WRITELN(P , ST); prints the message string on the page. Next, WRITELN(P); prints a blank line and Write (P , I:3); in the For loop prints the integers 1-10, allowing three print positions for each on the page, just as it would have on the screen if the file identification were not included in the WRITELN statement. The final WRITELN sends a return (ASCII 13) to the printer to cause it to print its internal buffer.

No Frills?

Frills control could be considered now that I had a reasonable way to send data to the printer. I wrote a Pascal procedure that allows me to use many of the Epson MX-80's special features from Pascal. That was something Applesoft never really let me do easily.

The program PRINTDEMO incorporates the printer control procedure I have written (called EPSON, for obvious reasons). This procedure requires a global variable, P, to be declared the type Text, and the type EPSONCOM to be defined as the set of commands that the procedure will respond to. This command list may be modified as desired to fit your application.

To open the file P so that it may be

Listing 1. Program to demonstrate Epson printer control procedure.

```
(*****************************************************************************)
(*                                                                           *)
(*                            PROGRAM PRINTDEMO;                             *)
(*                                                                           *)
(*        PROGRAM TO DEMONSTRATE EPSON PRINTER CONTROL PROCEDURE             *)
(*                                                                           *)
(*                              by FRED JOHNSON                              *)
(*                                                                           *)
(*****************************************************************************)

TYPE EPSONCOM = (ON,OFF,RES,ITAL,NORM,
                 EMPH,DSTR,SSTR,FF,
                 C40,C64,C80,C132);      (* EPSON PRINTER COMMANDS          *)

     COLUMN = C40..C132;                 (* NUMBER OF PRINT COLLUMNS ALLOWED *)

VAR  P : TEXT;                           (* USED TO OUTPUT TEXT TO PRINTER   *)


PROCEDURE EPSON( COMMAND : EPSONCOM );

VAR ESC : CHAR;

BEGIN
   ESC := CHR(27);
   CASE COMMAND OF
      ON   : REWRITE(P,'PRINTER:');      (* OPENS P AS TEXT FILE FOR PRINTER *)
      OFF  : CLOSE( P );                 (* CLOSES P                         *)
      RES  : WRITE(P,ESC,'@');           (* RESETS PRINTER TO TURN ON STATE  *)
      ITAL : WRITE(P,ESC,'4');           (* ALTERNATE CHARACTER SET          *)
      NORM : WRITE(P,ESC,'5');           (* NORMAL CHARACTER SET             *)
      EMPH : WRITE(P,ESC,'E');           (* EMPHASIZED PRINT                 *)
      DSTR : WRITE(P,ESC,'G');           (* DOUBLESTRIKE PRINT               *)
      SSTR : WRITE(P,ESC,'F',ESC,'H');   (* SINGLESTRIKE -- EMPH AND DSTR OFF *)
      C40  : WRITE(P,CHR(18),CHR(14));   (* 40 COLUMN  (EXPANDED)            *)
      C64  : WRITE(P,CHR(15),CHR(14));   (* 64 COLUMN  (COMP/EXP)            *)
      C80  : WRITE(P,CHR(18),CHR(20));   (* 80 COLUMN  (NORMAL TYPE)         *)
      C132 : WRITE(P,CHR(20),CHR(15));   (* 132 COLUMN (COMPRESSED)          *)
      FF   : WRITE(P,CHR(12));           (* FORM FEED                        *)
   END; (* CASE *)
END;  (* EPSON *)

PROCEDURE CENT( COL: COLUMN ; OUT: STRING );
                                         (* CENTERS OUTPUT AS COL COLLUMNS   *)
VAR TAB , WIDTH : INTEGER;               (* ON CURRENT LINE OF PRINTER       *)

BEGIN                                    (* WIDTH IS NUMBER OF PRINT COLUMNS *)
   CASE COL OF
      C40  : WIDTH := 40 ;               (* IF 64 OR 132 COLUMNS ARE USED    *)
      C64  : WIDTH := 64 ;               (* WHILE EMPHASIZED OR DOUBLESTRIKE *)
      C80  : WIDTH := 80 ;               (* ARE SELECTED THEN PRINTER WILL   *)
      C132 : WIDTH := 132;               (* REVERT TO 40 OR 80 COLUMN MODE   *)
   END;  (* CASE *)
   EPSON( COL );
   TAB := (WIDTH + LENGTH(OUT)) DIV 2;   (* TAB = RIGHT EDGE OF TEXT         *)
   WRITELN(P, OUT:TAB );
END;  (* CENT *)

PROCEDURE ZERO;

BEGIN
   PAGE(OUTPUT);                         (* CLEAR SCREEN *)
   GOTOXY( 10 , 11 );
   WRITE('PRINTER NOT ON LINE');         (* NOTE ON SCREEN TO TURN ON PRINTER *)
   GOTOXY( 6 , 11 );

   EPSON( ON );                          (* CLEAR 'PRINTER ON' NOTICE LEFT ON *)
   EPSON( RES );                         (* SCREEN ABOVE AS SOON AS PRINTER   *)
   WRITE('EPSON PRINTER DEMONSTRATION');(* ACCEPTS CHARACTER FROM COMPUTER    *)
   GOTOXY( 0,0 );
   EPSON( EMPH );                        (* DOUBLESTRIKE AND EMPHASIZED       *)
   EPSON( DSTR );                        (* TITLE CENTERED ON 40 COLUMN PAGE  *)
   CENT(C40,'UCSD / EPSON MX-80 DRIVER');
   EPSON( SSTR );                        (* RETURN TO SINGLE SINGLESTRIKE     *)
END;  (* ZERO *)

PROCEDURE ONE;

BEGIN
   WRITELN(P );
   WRITE(P, 'THIS PRINTER DRIVER ');
   EPSON( C132 );
   EPSON( DSTR );                        (* COMPRESSED DOUBLESTRIKE           *)
   WRITE(P, '1');                        (* USED FOR FOOTNOTE                 *)
   EPSON( C80 );
   EPSON( SSTR );
   WRITE(P, ' PROCEDURE AS IMPLEMENTED ');
   WRITELN(P, 'IN UCSD PASCAL ON THE APPLE II');
   WRITELN(P );
   WRITE(P, 'ALLOWS SELECTION OF ');     (* THIS ALTERNATE CHARACTER SET IS   *)
   EPSON( ITAL );                        (* AVAILABLE WHEN GRAFTRAX80 ROMS    *)
   WRITE(P, 'ITALICS,');                 (* ARE INSTALLED IN THE EPSON MX-80  *)
   EPSON( NORM );
   EPSON( C132 );
   WRITE(P, ' CONDENSED, ');
   EPSON( C40 );
   WRITE(P, 'EXPANDED,');
   EPSON( C80 );
   WRITE(P, 'OR');
   EPSON( C64 );
```

More →

used to output text, for example, all that is required is to include this line in the program:

```
EPSON( ON );
```

To write to the printer in expanded (40-column) print, it is necessary to write the following into the main Pascal program:

```
EPSON( C40 );
WRITELN(P, 'This is 40 column');
```

A sample program listing, as well as the program's output, is included in order to demonstrate how the Pascal

Listing continued.

```
    WRITELN(P, ' CONDENSED/EXPANDED');
END;  (* ONE *)

PROCEDURE TWO;

BEGIN
    WRITELN(P);
    EPSON( EMPH );
    WRITE(P, 'EMPHASIZED IS AVAILABLE IN ');
    EPSON( C40 );
    WRITE(P, 'EXPANDED, ');
    EPSON( C80 );
    WRITELN(P, 'AND NORMAL TYPE FACES');

    WRITELN(P);
    EPSON( SSTR );
    EPSON( DSTR );
    WRITE(P, 'THE SAME IS TRUE OF THE ');
    EPSON( C40 );
    WRITELN(P, 'DOUBLESTRIKE TYPEFACE');

    WRITELN(P);
    EPSON( EMPH );
    WRITE(P, 'EMPHISIZED DOUBLESTRIKE PRODUCES THE ');
    WRITELN(P, 'FINEST TYPE AVAILABLE IN DOT MATRIX PRINT');
END;  (* TWO *)

PROCEDURE THREE;

BEGIN
    WRITELN(P);                                  (* EXPANDED DOUBLESTRIKE ITALICS    *)
    EPSON( ITAL );
    CENT(C80,'IMPRESSIVE, ISN''T IT!');

    WRITELN(P );
    EPSON( NORM );
    EPSON( SSTR );
    WRITE(P, 'BOLD FACE TITLE LINES AND FOOTNOTES ');
    EPSON( C132 );
    EPSON( DSTR );
    WRITE(P, '2.');
    EPSON( C80 );
    EPSON( SSTR );
    WRITELN(P, ' WITH A WITH A PRINTED LOOK BECOME EASY ');

    WRITELN(P);
    WRITE(P, 'AND ALL THIS CONTROL COMES TO YOU IN ONE ');
    WRITELN(P, 'LITTLE PASCAL SUBROUTINE BUT AS');
END;  (* THREE *)

PROCEDURE FOUR;                                  (* SINGLE STRIKE NON EMPHASIZED     *)
                                                 (* TYPE GIVES FASEST PRINTING SPEED *)
BEGIN
    WRITELN(P);
    WRITE(P, 'YOU CAN SEE IF YOU WERE WATCHING THE ');
    WRITELN(P, 'PRINTER, YOU WILL PAY IN LOST SPEED');

    WRITELN(P);
    WRITE(P, 'FOR ALL THE EXTRA FRILLS AND FEATURES');
    WRITELN(P, ' IN THE PRINTED OUTPUT OF THIS PROGRAM.');

    WRITELN(P);
    EPSON( ITAL );
    EPSON( C132 );
    WRITE(P, '1. THIS EPSON DRIVER PROCEDURE MAY BE MODIFIED');
    WRITELN(P, ' FOR ANY PRINTER''S SET OF CONTROL CHARACTERS');
    WRITE(P, '2. THIS MIGHT BE USEFULL FOR FOOTNOTES ALTHOUGH');
    WRITELN(P, ' IT IS  A BIT AWKWARD TO USE FOR LARGE DOCUMENTS.');
END;  (* FOUR *)

PROCEDURE FIVE;

BEGIN
    EPSON( NORM );
    EPSON( SSTR );                               (* RETURN PRINTER TO NORMAL SETUP   *)
    EPSON( C80 );
    EPSON( FF );                                 (* FORM FEED                        *)
    PAGE( OUTPUT );
END;  (*  FIVE  *)


BEGIN    (* MAIN PROGRAM *)
    ZERO;
    ONE;                                         (* SIMPLE PROGRAM TO CALL PRINTING  *)
    TWO;                                         (* PROCEDURES IS NECESSARY TO ALLOW *)
    THREE;                                       (* ALL OF THESE WRITELN'S TO BE     *)
    FOUR;                                        (* COMPILED IN AVAILABLE MEMORY     *)
    FIVE;
END.
```

procedure is used to control the Epson printer. A second useful procedure, called Cent, is included in this program to center a string on the printed page.

Numerous comments have been included in the Pascal procedures to explain what is being done in each statement.

Since Pascal is compiled, these comments do not slow the execution of this program as they would in an interpreted language. Since the Pascal compiler would not allow all of the statements to be compiled in one procedure, the main program calls six procedures, labeled 0–5. Each of these is small enough to be compiled easily, and most of them will fit on the Apple's screen to aid in editing the program.

Procedure 0 writes both to the screen and to the printer. A note that says the printer is not on-line is put on the screen, and then a reset command is sent to the printer. If the printer is not on-line, the printer control card in slot 1 will not let the Apple continue with the program.

As soon as the printer command is accepted, the computer is allowed to continue. The message on the screen is replaced by a second message stating that the printer demonstration is in progress. If the printer is on-line when the program is first run, the first message is written over so quickly by the second it appears that the first message was never written to the screen.

Procedure 0 then sends emphasized and double-strike commands to the printer via the Epson procedure, and a 40-column title string is sent via the Cent procedure.

Procedures 1–4 demonstrate various printer commands and typefaces available from the Epson MX-80 printer. Procedure 5 returns the printer to its initial setup, issues a form feed to get a clean sheet of paper under the printhead and finishes off by clearing the video screen with a PAGE(OUTPUT); statement.

Quirk Quibbles

In using these procedures, two quirks of the Epson MX-80 should be noted.

First, the printer will not accept the 132- or 64-column modes of operation if emphasized or double-strike options are in effect. The printer will print 80 and 40 columns, respectively, if 132 or 64 are selected under these conditions.

The second quirk is that the printer forgets that it is in 40- or 63-column mode every time it receives a return character. This means that every WRITELN puts it back in 80- or 132-column mode. If you want a whole page of expanded (40-column) type, then you must include the command EPSON(C40); at the beginning of each line.

After writing this program, I can see why professional programmers like Pascal so much. Pascal's real beauty lies in the ease with which a program like this can be read, understood and modified by someone other than the original programmer.

By simply changing the characters each command sends to the printer, this procedure, or one similar to it, could make most printers do any of the tricks they are capable of.

After I figured out how the MX-80 and Apple's UCSD Pascal work together, I found that they make an extremely compatible couple.■

```
UCSD / EPSON MX-80 DRIVER

THIS PRINTER DRIVER 1 PROCEDURE AS IMPLEMENTED IN UCSD PASCAL ON THE APPLE II
ALLOWS SELECTION OF ITALICS, CONDENSED, EXPANDED, OR CONDENSED/EXPANDED
EMPHASIZED IS AVAILABLE IN EXPANDED, AND NORMAL TYPE FACES
THE SAME IS TRUE OF THE DOUBLESTRIKE TYPEFACE
EMPHISIZED DOUBLESTRIKE PRODUCES THE FINEST TYPE AVAILABLE IN DOT MATRIX PRINT
                    IMPRESSIVE, ISN'T IT!
BOLD FACE TITLE LINES AND FOOTNOTES 2. WITH A WITH A PRINTED LOOK BECOME EASY
AND ALL THIS CONTROL COMES TO YOU IN ONE LITTLE PASCAL SUBROUTINE BUT AS
YOU CAN SEE IF YOU WERE WATCHING THE PRINTER, YOU WILL PAY IN LOST SPEED
FOR ALL THE EXTRA FRILLS AND FEATURES IN THE PRINTED OUTPUT OF THIS PROGRAM.

1. THIS EPSON DRIVER PROCEDURE MAY BE MODIFIED FOR ANY PRINTER'S SET OF CONTROL CHARACTERS
2. THIS MIGHT BE USEFULL FOR FOOTNOTES ALTHOUGH IT IS A BIT AWKWARD TO USE FOR LARGE DOCUMENTS.
```

Example 1. Sample of UCSD/Epson MX-80 driver output.

Part 2

Smarter Than the Average Atari

In the wrap-up of this two part article, the authors explain the nuts and bolts of using your Atari as an intelligent peripheral.

By Brian and Christine Ence

Last month I suggested using your Atari 400 as an intelligent peripheral. There are two basic problems that need to be solved. The first problem is providing a method to save Atari programs; the second is finding a means for orderly communication between computers.

Part 1 described a way to save Atari programs with the disk drive of your computer using the Atari CSave and CLoad commands. This involves building an electronic interface and writing a program to make your computer emulate the Atari cassette recorder.

Part 2 describes one way to communicate with an Atari.

I include information on interrupt handling and programming the Atari Pokey chip for serial I/O; a description of how I use this information for communication between the computers; and a demonstration program using the Atari as a graphics peripheral.

I'll also suggest other applications of the Atari as an intelligent peripheral.

Address correspondence to Brian and Christine Ence, 1015 East Main St., Lansdale, PA 19446.

Communications Procedure

As I indicated earlier, communication between my computers is regulated by the control lines of the serial ports. When one computer needs an input from the other, it sends a pulse on a control line.

This pulse sets a flag at the other computer. Periodically, that computer checks the flag. If the flag has been set, that computer responds by returning a byte.

I chose this approach so that I can use Atari Basic to keep the programs simple and easy to modify. Sending one byte per pulse, I am assured of never missing a byte, regardless of how slow the Atari programs run.

To be more specific, my H89 sends a pulse on the CTS line of the RS-232C port to the Proceed input of the Atari. The Atari sends a pulse using its Command output line to the DTR input of the H89's serial port.

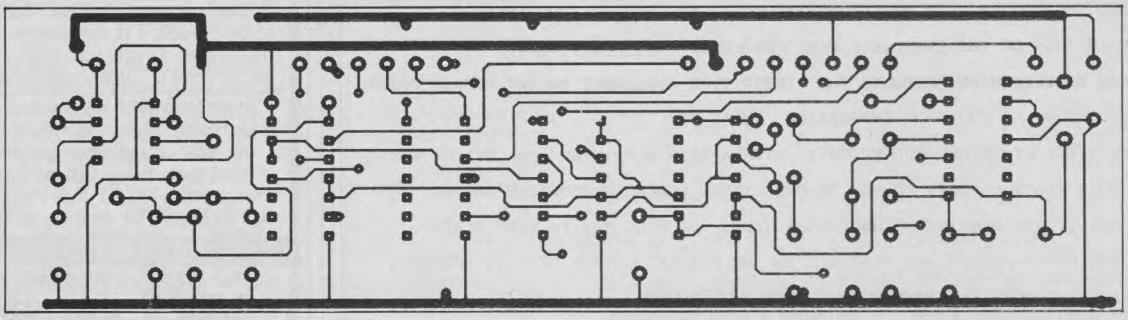


Fig. 1. Circuit board pattern as seen from the solder side.

Using the Command output this way will probably interfere with the use of any Atari peripheral attached to the serial port but, with some additional effort, this conflict can be avoided.

I have found that I can reliably detect the receipt of a pulse at my H89 by reading the modem-status register of my serial port. At the Atari, the pulse is received by a 6520 peripheral interface adapter.

This IC can be programmed to set a bit in one of its registers when it receives a pulse and it should be possible to read this bit using a Peek statement.

I found, however, that I couldn't consistently detect a pulse this way, probably because of a conflict with the Atari operating system. My solution is to allow the pulse to generate an interrupt in the Atari (See the sidebar "IRQ interrupts in the Atari.") The interrupt handler sets a flag that can be easily read with a Peek statement.

Atari Initialization

Initializing the Atari to communicate with another computer takes three steps: programming first the serial port; second, the sound generators (which provide the timing for the port); and third, the interrupts. Information necessary for performing each of these steps is given in the sidebars.

Listing 1 illustrates the use of this information. This program initializes the Atari and then enables the Atari to act as a simple graphics peripheral.

The serial port is programmed in line 95; the sound generators in lines 110 to 120; and the interrupts in lines 140 to 230. The rest of the program performs I/O functions and executes the graphics commands.

I should warn you that after you alter the interrupt vectors in this initialization, the Atari cannot perform a CSave. If you want to save a program, you must first reset the Atari, an action that returns the interrupt vectors to their original values.

However, I have found that the first program saved after a system reset is missing the command byte of the first record. With this byte missing, the program won't load properly.

My solution has been to start a CSave and stop it with the break key after it has started sending data. The next CSave then loads properly.

The I/O functions in this program are performed in lines 1000–1150. The input routine, starting at line 1000, first sends a pulse on the command line of the serial port. This signals the host computer to send a byte.

When the byte has been received, a

Bit 6	Bit 5	Bit 4	Mode
0	0	0	Transmit and receive rates set by external clock.
0	0	1	Transmit rate set by external clock. Receive rate set by audio channel 4, asynchronous.
0	1	0	Transmit and receive rates set by channel 4.
1	0	0	Transmit rate set by channel 4. Receive rate set by external clock.
1	1	0	Transmit rate set by audio channel 2. Receive rate set by audio channel 4.
1	1	1	Same as 110, but receive is asynchronous.
0	1	1	Not useful.
1	0	1	Not useful.

Table 1. Definitions of serial port modes.

Parts List

R1	10K Variable	C1 .1 μF	IC1 XR2211
R2	15K	C2 .047 μF	IC2 CD4093
R3	68K	C3 10 μF	IC3 CD4001
R4	100K	C4 10 μF	IC4 MC1489
R5	560K	C5 .1 μF	IC5 MC1488
R6	4.7K	C6 .01 μF	IC6 LM555
R7	1K	C7 .1 μF	
R8	1K	C8 .01 μF	D1 1N914
R9	10K	C9 .1 μF	D2 1N914
R10	1K	C10 .0022 μF	(or equivalent)
		C11 .0022 μF	
		C12 .1 μF	LED any miniature LED
		C13 .1 μF	

All resistors are 1/8 watt 10 percent.
C3 and C4 should be miniature electrolytics.
All other capacitors may be mylar or ceramic disk.

Table 2. Parts list.

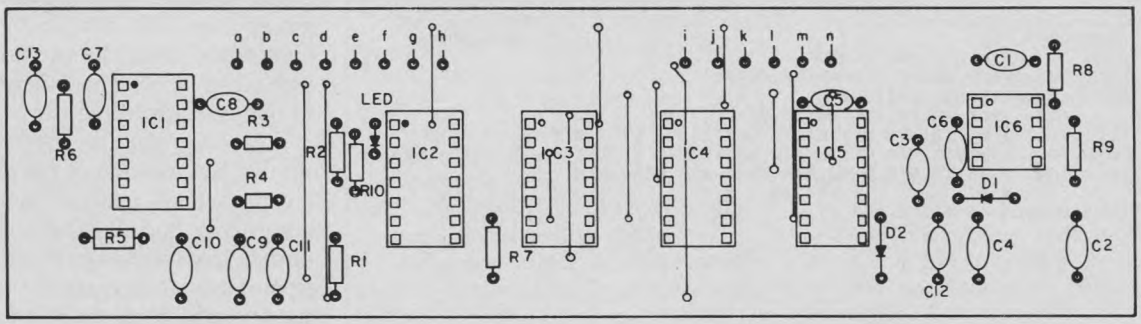


Fig. 2. Circuit board components and jumper replacements.

serial-in-ready interrupt occurs and the handler sets the byte at address 1791 to 255.

When the program senses that the data has been received by finding the byte at 1791 not equal to zero, it sets the variable in equal to the contents of the serial-in register. Location 1791 is then set to zero in preparation for another input, and the subroutine is exited.

For output from the Atari, the host first sends a pulse on the Proceed line. This pulse generates an interrupt that sets the byte at 1775 to 255, thereby signaling the Atari that the host is ready for data.

The byte at 1790 is used as a flag that equals 255 if the serial-out register is empty and ready to send another byte. When the register is ready, the contents of variable Out is placed in it, causing that byte to be sent; the subroutine is then exited.

```
90 REM INITIALIZE SERIAL CONTROL
91 REM RECIEVE USING CHANNEL 2;SEND
 USING 4, ASYNCHRONOUS
95 POKE 53775,115:POKE 562,115
100 REM INIT. AUDIO SECTION
105 REM SET CHANNELS 2 & 4 TO HIGH
 PRECISION, 19,200 HZ
110 FOR I=0 TO 8:READ A:POKE 53760+I,A:N
EXT I
120 DATA 40,160,0,160,40,160,0,160,120
130 REM LOAD INTERRUPT HANDLERS
140 FOR I=0 TO 23:READ A:POKE 1768+I,A:N
EXT I
145 DATA 169,255,141,239,6,104,64,0
150 DATA 169,255,141,255,6,104,64
160 DATA 169,255,141,254,6,104,64,255,0
170 REM SET INTERRUPT VECTORS
175 POKE 514,232:POKE 515,6
180 POKE 522,240:POKE 523,6
190 POKE 524,247:POKE 525,6
200 REM INITIALIZE INTERRUPTS
201 REM ENABLE: BREAK KEY, OTHER KEYS
 SERIAL IN AND SERIAL,OUT
210 POKE 16,240:POKE 53774,240
220 REM ENABLE INTERRUPT ON RISING
  PROCEED INPUT
230 POKE 54018,63
500 REM GRAPHICS INTERPRETER
505 GRAPHICS 8:COLOR 1
509 REM GET COMMAND # AND GO TO COMMAND
EXECUTION SUBROUTINE
510 GOSUB 1000
520 ON IN GOSUB 2000,2100,2200,2300,2400
,2500
530 GOTO 510
1000 REM INPUT ROUTINE
1010 POKE 54019,60:POKE 54019,52
1020 IF PEEK(1791)=0 THEN 1020
1030 IN=PEEK(53773)
1040 POKE 1791,0
1050 RETURN
1100 REM OUTPUT ROUTINE
1110 IF PEEK(1775)=0 THEN 1110
1115 POKE 1775,0
1120 IF PEEK(1790)=0 THEN 1120
1130 POKE 1790,0
1140 POKE 53773,OUT
1150 RETURN
2000 REM PLOT SUBROUTINE
2010 GOSUB 1000:Y=IN
2020 GOSUB 1000:XHI=IN
2030 GOSUB 1000:XLO=IN
2040 X=256*XHI+XLO
2050 PLOT X,Y:RETURN
2100 REM DRAW SUBROUTINE
2110 GOSUB 1000:Y=IN
2120 GOSUB 1000:XHI=IN
2130 GOSUB 1000:XLO=IN
2140 X=256*XHI+XLO
2150 DRAWTO X,Y:RETURN
2200 REM RELATIVE PLOT ROUTINE
2210 GOSUB 1000:YINC=IN
2220 IF YINC>127 THEN YINC=YINC-256
2230 Y=Y+YINC
2240 GOSUB 1000:XINC=IN
2250 IF XINC>127 THEN XINC=XINC-256
2260 X=X+XINC
2270 IF Y>191 THEN Y=191:IF Y<0 THEN Y=0
2280 IF X>319 THEN X=319:IF X<0 THEN X=0
2290 PLOT X,Y:RETURN
2300 REM RELATIVE DRAW ROUTINE
2310 GOSUB 1000:YINC=IN
2320 IF YINC>127 THEN YINC=YINC-256
2330 Y=Y+YINC
2340 GOSUB 1000:XINC=IN
2350 IF XINC>127 THEN XINC=XINC-256
2360 X=X+XINC
2370 IF Y>191 THEN Y=191:IF Y<0 THEN Y=0
2380 IF X>319 THEN X=319:IF X<0 THEN X=0
2390 DRAWTO X,Y:RETURN
2400 REM LOCATION ROUTINE
2410 OUT=Y:GOSUB 1100
2420 XHI=INT(X/256):XLO=X-256*XHI
2430 OUT=XLO:GOSUB 1100
2440 OUT=XHI:GOSUB 1100
2450 RETURN
2500 REM CLEARSCREEN ROUTINE
2510 GRAPHICS 24:RETURN
```

Listing 1. Demonstration of Atari graphics.

```
SCR # 1
  0 \ DEMOGRAF  INIT.PORT READ.ATR WRITE.ATR                    23JUL83BME
  1 208 CONSTANT PORT 6 CONSTANT LSD 0 CONSTANT MSD  \  FOR 19200 Bd
  2 : INIT.PORT    0 PORT 3 + P! 0 PORT 1 + P!  \ DISABLE INTERRUPTS
  3      16 PORT 4 + P!                        \ SET LOOP BACK
  4      128 PORT 3 + P! LSD PORT P! MSD PORT 1 + P! \ SET BAUD
  5      3 PORT 3 + P!                  \ 8 BITS, NOPARITY, 1STOP BIT
  6      PORT P@ DROP                   \ READ ONCE
  7      34 MS PORT P@ DROP             \ WAIT, READ AGAIN
  8      0 PORT 4 + P!                  \ CLEAR LOOP BACK
  9      PORT 6 + P@ DROP ;             \ CLEAR MODEM STATUS REG.
 10 : READ.PORT    BEGIN PORT 5 + P@ 1 AND UNTIL PORT P@ ;
 11 : WRITE.ATR    BEGIN PORT 6 + P@ 17 AND 17 XOR
 12      0= UNTIL PORT P! ;
 13 : REQ.IN   PORT 4 + DUP 2 SWAP P! 0 SWAP P! ;
 14 : READ.ATR  REQ.IN READ.PORT ;
 15 -->

SCR # 2
  0 \ DEMOGRAF                                                 10JUL83BME
  1 : PLOT   1 WRITE.ATR WRITE.ATR
  2         256 /MOD WRITE.ATR WRITE.ATR ;          ( X Y ... )
  3 : DRAW   2 WRITE.ATR WRITE.ATR
  4         256 /MOD WRITE.ATR WRITE.ATR ;          ( X Y ... )
  5 : PLOT.INC   3 WRITE.ATR WRITE.ATR WRITE.ATR ;  ( X Y ... )
  6 : DRAW.INC   4 WRITE.ATR WRITE.ATR WRITE.ATR ;
  7 : LOCATION   5 WRITE.ATR READ.ATR READ.ATR READ.ATR
  8         256 * + . . ;     ( PRINTS X Y )
  9 : CLEARSCREEN   6 WRITE.ATR ;
 10 ;S
 11
 12
 13
 14
 15
```

Listing 2. Heath program for demonstrating Atari graphics.

Graphics Commands

Each graphics command consists of a command number and a set of data. The number of the command is sent to the Atari, where it is used to select a subroutine to execute the command (lines 500 to 530).

The selected subroutine then accepts the data that follows the command number and uses it to execute the command. These subroutines begin at line 2000.

The first two commands are Plot and Draw. They begin at lines 2000 and 2100, respectively, and perform built-in commands of the Atari. Note that the highest resolution graphics mode is being used and, as a result, the value of the x variable may be as high as 319.

Since the highest number that can be sent in one byte is 255, I have to break the x value down to high and low bytes, send them separately and compute the value of x at the Atari.

Relative-plot and relative-draw commands are performed by the routines at lines 2200 and 2300. These commands aren't inherent in the Atari and illustrate how new graphics commands can be created.

In these commands, the plot, or draw-to, location is given relative to the current location of the cursor. The data is sent in two's complement notation to allow positive and negative values to be sent in a single byte. These commands make it much easier to draw a given figure repeatedly anywhere on the screen.

The last two commands, which begin at lines 2400 and 2500, locate the cursor and clear the screen. The first command sends the current location of the cursor to the host.

This command is useful in determining the location of the cursor after a number of relative plot or draw commands and illustrates the sending of data from the Atari to the host. The final command clears the screen and uses no additional data from the host.

What's Your Function?

After you have studied this graphics

program, you can guess which functions are performed by the host. Listing 2 is a program in HForth-79 that provides these functions on my H89. Together these two programs let me use the Atari as a graphics peripheral.

The communications procedures at the H89 are essentially the same as for the Atari. The details are executed in the words READ.ATR and WRITE-.ATR.

These words are written for use with an INS8250 asynchronous communications IC, which controls the serial port of my H89. Your program will have to perform these same functions.

READ.ATR first requests an input (REQ.IN) by sending a pulse on the CTS line and waits for a byte to be received (READ.PORT). WRITE.ATR first waits for a pulse to be received on the RTS line and then writes the byte to the transmitter holding register.

More Applications

These simple programs use only a fraction of the capabilities of the Atari. You can write programs that make all of your Atari's color graphics modes and sound functions available to your computer.

You can write programs to make all of the Atari's color graphics modes available to your computer.

Special purpose commands can be programmed that, for example, draw and fill in a box, given the coordinates of two opposite corners. This can make the plotting of bar graphs and histograms simple.

Commands for circles and arcs can also be added. I have written programs for doing turtle graphics using the Atari P/M graphics in which I send commands to the Atari as if it were the turtle.

Going beyond the graphics applications, the joystick and paddle ports can be read and sent to your computer, giving it parallel inputs and some measure of A to D conversion capability. You can also use the joystick ports as outputs for control applications.

A printer with a parallel interface can be operated through the joystick ports. With a printer hooked up in this way, and using assembly-language programming, the Atari can serve as an intelligent buffer that holds several pages of text.

Another way I intend to use the Atari is to provide floating-point arithmetic functions when I'm using HForth-79 on my computer.

You may find that you need different communication procedures for different uses of the Atari. For example, you may want to add a data buffer at either computer so that more than one byte can be sent at one time.

The control lines can then be used to indicate a full buffer. The information I have given in the boxes on interrupts, sound generators and control of the serial port will allow you to develop a communications system to suit your purpose.■

IRQ Interrupts in the Atari

IRQ interrupts are interrupts that are sensed by the IRQ input of the 6502 CPU in the Atari. These may be masked or blocked from affecting the operation of the CPU by setting a mask bit in one of the registers in the CPU.

IRQ interrupts are generated either by the Atari Pokey IC or by the 6520 PIA. While all IRQ interrupts can be masked by setting a single bit in the CPU, they may also be individually masked in the Pokey and the PIA.

Most interrupts are enabled by setting the appropriate bits in the IRQEN register in the Pokey. This register is located at 53774 and its shadow register is at 16. The bit to be set for each possible source of interrupts is given below:

Bit 7: Break key (break key is depressed).
Bit 6: Other key (any other key is depressed).
Bit 5: Serial-in ready (the serial input register is full and may now be read).
Bit 4: Serial-out ready (data may be stored in the SEROUT register).
Bit 3: Transmission finished (the output shift register is empty).
Bit 2: Timer #4 (audio divider #4 has timed out).
Bit 1: Timer #2 (audio divider #2 has timed out).
Bit 0: Timer #1 (audio divider #1 has timed out).

The Proceed and Interrupt inputs of the Atari serial port are handled by the PIA. Proceed is controlled by bits 1 and 0 of the PACTL register at address 54018; Interrupt, by bits 1 and 0 of PBCTL at 54019.

For each input, setting bit 0 of the corresponding register enables the interrupt. If bit 1 is set (1), the interrupt occurs on the leading edge of the input signal. If bit 1 is clear (0), the interrupt is generated on the falling edge. (For those familiar with the PIA, Proceed is the CA1 input; Interrupt, the CB1 input; Command, the CB2 output. The CA2 output drives a transistor for the motor output.)

When an enabled IRQ interrupt occurs, it causes program control to transfer to the system IRQ handler routine. This routine determines the source of the interrupt by examining IRQST (the IRQ Status register at 53774 in the Pokey), PACTL and PBCTL.

It then jumps to the appropriate routine to handle the specific interrupt. The addresses for these routines are stored in memory.

If you wish to provide your own interrupt-handling routines, put the address of your routine in place of the one provided by the operating system. The locations in memory where the handler addresses are stored include:

Memory Address	Source of interrupt
514	Proceed input
516	Interrupt input
518	Break key depressed
520	Any other key depressed
522	Serial-in data ready
524	Serial-out data ready
526	Transmission finished
528	Pokey timer #1 timed out
530	Pokey timer #2 timed out
532	Pokey timer #4 timed out

Because of the way that the 6502 uses this data, the handler addresses are stored as two bytes, beginning at the address given above with the low-order byte first.

If you want to provide your own interrupt handler, you should know that the system interrupt handler pushes the accumulator onto the stack before turning control over to the specific interrupt handler. Your handler should end by pulling the accumulator (PLA) and executing a return from interrupt (RTI).□

B.E.

Serial Port Control

The operation of the serial port is controlled by setting bits in the register SKCTL at address 53775 (SKCTL has a shadow register, SSKCTL at 562.) The following functions can be performed by setting the bits as indicated below:

Bit 7: Break mode: force serial output to zero.

Bit 6:
Bit 5:
Bit 4: } Serial port mode: see the chart below for available options.

Bit 3: FSK mode: serial data is transmitted as high and low tones instead of high and low voltages. (It uses audio channels 1 and 2 to provide the tones.)

Bit 2: Fast Pot Scan: a lower-resolution, high-speed scan of the paddle inputs.

Bit 1: Enable key scan.

Bit 0: Enable key debounce.

(Bits 1 and 0 set to 0 gives a testing and initialization state.)

The Serial Port Modes are defined in Table 1.

When receiving in the asynchronous mode, audio channels 3 and 4 are zeroed at the beginning of each start bit. When an external clock is used, it is at the data rate (not 16 times the data rate as is usual). Data changes when the clock goes high and is sampled when it goes low.

The FSK mode cannot be used for modes 110 or 111 since audio channel 2 is used for the transmit data rate and is unavailable for providing on of the two tones.□

B.E.

Connections to the Circuit Board:

Connections to the Atari serial port:

a Atari data out
b Motor
c +5 volts
d Ground
e Atari data in
f Proceed
g +12 volts
h Command

Connections to the RS-232C port:

i Ground
j CTS
k Computer data out
l Computer data in
m DTR
n RTS

Sound Generators and Serial Port Timing

The sound generators in the Pokey not only produce sound effects but may also provide the timing for the reception and transmission of data in the serial port and the tones used for recording data on the cassette recorder. (See the sidebar on serial port control.)

If bit 3 of SKCTL is set, the tones produced by sound generators 1 and 2 are transmitted whenever a 1 or a 0, respectively, is to be sent from the serial port.

For some settings of SKCTL, the data rate for sending or receiving are set by sound generators 2 and 4. Typically, when sound generators 2 and 4 are used for setting the data rate, the sound generators are put in a high-precision mode.

This is done by setting bits 3, 4, 5, and 6 in register AUDCTL at 537680. The following routines program the sound generators to the desired data rate (RATE).

```
10 N=INT(1789790/(2*RATE)-7+.5)
20 MSB=INT(N/256):LSB=N-256*MSB
```

To set channel 2 to the high-precision mode:

```
30 POKE 53768,80
40 POKE 53760,LSB:POKE 53762,MSB
50 POKE 53761,160:POKE 53763,160
```

To set channel 4 to the high-precision mode:

```
30 POKE 53768,40
60 POKE 53764,LSB:POKE 53766,MSB
70 POKE 53765,160:POKE 53767,160
```

To set both channels to the high-precision mode, change line 30 to:

```
30 POKE 53768,120
```

and if desired, recompute MSB and LSB between lines 50 and 60.□

B.E.

Construction Tips

If the circuit board pattern given in Fig. 1 is used, the holes should be drilled with a number 67 drill.

To avoid solder bridges, use a 25-watt soldering iron with a sharp pointed tip and don't let solder accumulate on the tip.

Insert and solder the 14 jumpers first, then the IC sockets and, finally, the discrete components.

If you cannot find a 10K variable resistor for R1, use a 10K miniature trimpot with the terminal for the slider connected to one of the other terminals.

When you have finished soldering, inspect the board carefully to be sure that there are no solder bridges.

No mounting holes were provided on the circuit pattern. The board is sized to fit into a Radio Shack project case, cat. no. 270-223, which has internal slots to hold circuit boards.

An etched and drilled circuit board is available for $9 from AP Services, PO Box 1332, North Wales, PA 19454.

For H89 users, a disk with the following contents is available for $15 from the same source:

- A more sophisticated recorder emulator program that allows disk to be changed, their catalog to be viewed and files to be deleted. Both compiled object code and source code in HForth-79 are provided.
- A printer emulator program that allows you to list Atari programs to your H89 for viewing on your console or for storing to disk to be printed out at a later time. Again both source code and compiled object code are provided.
- Atari Basic code and HForth-79 source code for a turtle graphics program.
- Complete documentation for all of the programs. These programs run under HDOS 2.0 and are on a ten-sector hard disk.□

B.E.

NEW FROM

Wayne Green Books

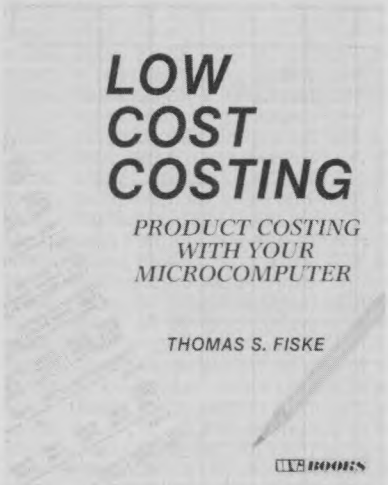
CONTROL YOUR COSTS AND MAXIMIZE YOUR PROFIT

To make a profit, you have to know what your costs are. *Low Cost Costing* is a book-and-software combination that will help you determine production costs for your small business, using your Apple, IBM PC, or TRS-80.

Whether you're a company president, controller, production manager, or anyone responsible for production, you can increase your profits once you know what you're spending. *Low Cost Costing* shows you how to analyze your costs using your profit-and-loss statements and your microcomputer. Once you've found them, you can:

Cut your costs.
Adjust prices for maximum profit.
Set marketing and manufacturing strategies.
Predict the results of changes in volume and fixed or variable costs.
Determine why some products sell and others don't.

The *Low Cost Costing* book and disk are packaged in a plastic binder that stores and protects them. It's the time- and money-saving combination your small business needs—*Low Cost Costing* and your micro.



Low Cost Costing
Thomas S. Fiske
approx. 112 pp.
$24.97

Apple II, II Plus, IIe
BK7399
ISBN 0-88006-067-0

IBM PC
BK7402
ISBN 0-88006-071-9

TRS-80 Model I, III
BK7403
ISBN 0-88006-072-7

To order, call toll-free 1-800-258-5473 for credit card orders. Or mail your order with check or money order or complete credit card information to: Wayne Green Books, Retail Sales, Peterborough, NH 03458. Please include shipping and handling of $1.50 for the first system, $1.00 for each additional system.

Yes, I want *Low Cost Costing*. $24.97 per system, plus shipping and handling ($1.50 for the first system, $1.00 for each additional system). Please indicate total number wanted for each system.

_____ Apple (BK7399) _____ IBM PC (BK7402) _____ TRS-80 (BK7403)

☐ payment enclosed ☐ MasterCard ☐ Visa ☐ Am. Ex.

Card # _____ MC bank # _____
Expiration date _____ Signature _____
Name _____
Address _____
City _____ State _____ Zip _____
Wayne Green Books, Peterborough, NH 03458 343B7L

Draw Your C-64 Out of Its Shell

Add nine turtle graphics commands to the Commodore-64's repertoire.

By Richard Holleran

Listing 1. Assembly language listing of the turtle graphics routines.

```
00001  0000              ; TURTLE GRAPHICS FOR THE 64
00002  0000              ; AUTHOR: RICHARD HOLLERAN
00003  0000              ;
00004  0000              ;
00005  0000              ;
00006  0000              DX     = $8000          ; CHANGE IN X DIRECTION
00007  0000              DY     = $8005          ; DHANGE IN Y DIRECTION
00008  0000              XCOR   = $800A          ; CURRENT X COORDINATE
00009  0000              YCOR   = $800F          ; CURRENT Y COORDINATE
00010  0000              CURANG = $8014          ; CURRENT HEADING OF TURTLE
00011  0000              STARTX = $8019          ; INITIAL X VALUE (159)
00012  0000              STARTY = $801E          ; INITIAL Y VALUE (99)
00013  0000              STARTA = $8023          ; INITIAL HEADING (3*PI/2)
00014  0000              DEGRAD = $8028          ; PI/180 FOR RADIAN CONVERSION
00015  0000              SQUARE = $802D          ; VALUE TO 'SQUARE' IMAGE
00016  0000              COLOR  = $8061          ; ENTRY POINT USED BY 'DRAW'
00017  0000              FINDDX = $8085          ; CALCULATE DX
00018  0000              FINDDY = $80A4          ; CALCULATE DY
00019  0000              PLOT1  = $80C7          ; ENTRY POINT USED BY 'MOVE'
00020  0000              IGR0   = $8047          ; GR#0 ENTRY POINT
00021  0000              IGR1   = $803F          ; GR#1 ENTRY POINT
00022  0000              FLAG   =$97             ; TAIL UP/DOWN FLAG (0=UP)
00023  0000              TABLE  = $8227          ; KEYWORD TABLE
00024  0000              BITVAL = $821F          ; TABLE OF BIT VALUES
00025  0000              CHRGET = $0073          ; ROUTINE TO GET NEXT CHAR
00026  0000              IERROR = $A43B          ; BASIC ERROR ROUTINE
00027  0000              READY  = $A474          ; PRINT 'READY.'
00028  0000              LET    = $A9A5          ; BASIC'S 'LET' ROUTINE
00029  0000              EVALEX = $AD9E          ; INPUT + EVALUATE EXPRESSION
00030  0000              FIXFLO = $B391          ; CHANGE REAL NUMBER TO INTEGER
00031  0000              INPUT2 = $B7EB          ; INPUT TWO VALUES
00032  0000              PLUS   = $B867          ; BASIC'S ADDITION ROUTINE
00033  0000              MULT   = $BA28          ; BASIC MULTIPLICATION ROUTINE
00034  0000              MEMFAC = $BBA2          ; MOVE MEMORY TO FACC#1
00035  0000              FACMEM = $BBD7          ; MOVE FACC#1 TO MEMORY
00036  0000              FLOFIX = $BC9B          ; CHANGE INTEGER TO REAL NUMBER
00037  0000              COSINE = $E264          ; BASIC COSINE EVALUATION
00038  0000              SINE   = $E26B          ; BASIC SINE EVALUATION
00039  0000              ;
00040  0000                    .MAC XLOYHI       ; PREPARE X AND Y REGISTERS
00041  0000                    LDX #<?1          ; LO BYTE OF ADDRESS IN X
00042  0000                    LDY #>?1          ; HI BYTE OF ADDRESS IN Y
00043  0000                    .MND
00044  0000              ;
00045  0000                    .MAC ALOYHI       ; PREPARE A AND Y REGISTERS
00046  0000                    LDA #<?1          ; LO BYTE OF ADDRESS IN A
00047  0000                    LDY #>?1          ; HI BYTE OF ADDRESS IN Y
00048  0000                    .MND
00049  0000              ;
00050  0000                    .MAC UPDATE       ; MOVE FACC#1 TO MEMORY
00051  0000                    XLOYHI ?1
00052  0000                    JSR FACMEM
00053  0000                    .MND
00054  0000              ;
00055  0000                    .MAC FETCH        ; MOVE MEMORY TO FACC#1
00056  0000                    ALOYHI ?1
00057  0000                    JSR MEMFAC
00058  0000                    .MND
00059  0000              ;
```

More →

There are two interesting aspects to my turtle graphics program: what it does and how it's implemented. This is a program that adds nine turtle graphics commands to the Commodore-64, including Draw, Plot, Move, Left and Right. When you run the program, it moves the machine language into upper memory, leaving about 21Kb of free memory (Listings 1 and 2).

As you probably know, turtle graphics is a popular computer graphics environment in which a "turtle" (either imaginary or graphically represented) is programmed to move around the screen, leaving a visible trail behind it. In my version the turtle is, unfortunately, imaginary.

A Commanding Look

The Draw command initializes the turtle's location and heading and clears the screen. Draw also sets the colors to a light blue foreground and a dark blue background.

GR# X lets you switch between the text screen (GR# 0) and the graphics screen (GR# 1). This is helpful if you want to review the graphics screen (you wouldn't use Draw, since that clears the screen). Any error condition that occurs while the graphics screen is being displayed causes a return to the text screen, which notifies you of the error.

COLR X,Y lets you set the screen- and line-drawing colors. For example, the COLR 6,14 command sets the col-

Address correspondence to Richard Holleran, 628 West St., Keene, NH 03431.

ors to light blue lines on a dark blue screen—the default settings. The color numbers are the same as those specified in your Commodore-64 manual.

The Move command moves the imaginary turtle, which leaves a visible trail. The syntax is MOVE X, where x is a positive number less than 511 or an expression with a result less than 511. A value larger than 510 produces a line of unpredictable length. A negative value has no effect on the display.

The Left X command alters the heading of the turtle by x degrees in a counterclockwise direction. Right X has a similar effect in a clockwise direction.

Plot X,Y lets you plot individual points. The x-axis range is 0–319; the y-axis range is 0–199. The point 0,0 is in the lower left-hand corner of the screen. Like the Move command, points outside of the visible range aren't plotted, which eliminates the possibility of poking a number into the middle of a running program. However, in the case of Plot, any y value outside the byte range (0–255) causes an illegal-quantity error.

The Tailup command lets you move the turtle without drawing a line: use TAILUP: MOVE 50, for example. The Taildown command sets the turtle's tail back down.

I would rather have used the standard Penup/Pendown commands, but Pendown contains the Basic keyword End so it can't be used.

Squaring

In addition to these commands, you can use the USR function to "square" the picture on your screen. Because of the shape of the pixels generated by the C-64, horizontal and vertical lines of equal magnitude don't appear to be of equal length. Try this example:

```
DRAW: FOR T#1T04: MOVE 90:
LEFT 90: NEXT
```

Depending on your monitor, this may or may not look like a square. On my monitor it appears square when the line in the y direction is three-fourths the magnitude of the x line. As a result, the default USR value is .75.

If the picture you see is too tall, substitute a smaller value, such as .65. If your picture is too wide, try a larger value, such as .8. Once you have found the proper value you need, set it only once when you load the program. The syntax of the USR function is A=USR(X), as in, for example,

Listing continued.

```
00060  0000                      .MAC ADD        ; ADD VALUE IN MEMORY TO FACC#1
00061  0000                      ALOYHI ?1
00062  0000                      JSR PLUS
00063  0000                      .MND
00064  0000                                      ;
00065  0000                      .MAC MULPLY     ; MULTIPLY FACC#1 -
00066  0000                      ALOYHI ?1       ; BY VALUE IN MEMORY
00067  0000                      JSR MULT
00068  0000                      .MND
00069  0000                                      ;
00070  0000                ; '?1' TAKES THE FIRST PARAMETER OF A MACRO CALL.
00071  0000                ; '#<' TAKES THE LOW BYTE, '#>' TAKES THE HIGH.
00072  0000                ;
00073  0000                *       = $0801
00074  0801                ;
00075  0801                ; PROGRAM INCLUDES A LINE OF BASIC FOR INITIALIZATION
00076  0801  0C            BASIC   .BYTE $0C,$08,$0A,$00,$9E,$20,$32
00076  0802  08
00076  0803  0A
00076  0804  00
00076  0805  9E
00076  0806  20
00076  0807  32
00077  0808  30                      .BYTE $30,$36,$32,$00,$00,$00 ; 10 SYS 2062
00077  0809  36
00077  080A  32
00077  080B  00
00077  080C  00
00077  080D  00
00078  080E                ;
00079  080E  A9 37         INIT    LDA #$37
00080  0810  85 01                STA $01         ; ENSURE BASIC ROM ENABLED
00081  0812  A9 A0                LDA #$A0
00082  0814  85 FC                STA $FC         ; SET UP POINTER TO BASIC ROM
00083  0816  A0 00                LDY #$00
00084  0818  84 FB                STY $FB
00085  081A  B1 FB         LOOP1   LDA ($FB),Y
00086  081C  91 FB                STA ($FB),Y     ; TRANSFER BASIC -
00087  081E  C8                   INY             ; TO RAM UNDERNEATH.
00088  081F  D0 F9                BNE LOOP1
00089  0821  E6 FC                INC $FC         ; WHEN $FC HOLDS #$C0 -
00090  0823  24 FC                BIT $FC         ; WHICH IS END OF BASIC -
00091  0825  50 F3                BVC LOOP1       ; THEN BRANCH WILL FAIL.
00092  0827  A9 36                LDA #$36
00093  0829  85 01                STA $01         ; ENABLE BASIC IN RAM
00094  082B  A9 81                LDA #$81        ; THIS IS HIGH BYTE FOR -
00095  082D  8D 01 03             STA $0301       ; ERROR VECTOR, -
00096  0830  8D 12 03             STA $0312       ; USR ROUTINE VECTOR, -
00097  0833  8D 06 A8             STA $A806       ; AND NEW PARSER.
00098  0836  A9 C4                LDA #$C4
00099  0838  8D 00 03             STA $0300       ; LOW BYTE OF ERROR VECTOR
00100  083B  A9 D2                LDA #$D2
00101  083D  8D 11 03             STA $0311       ; USR ROUTINE VECTOR LOW BYTE
00102  0840  A9 DC                LDA #$DC
00103  0842  8D 05 A8             STA $A805       ; NEW PARSER LOW BYTE
00104  0845  A9 80                LDA #$80
00105  0847  85 FE                STA $FE         ; SET UP POINTER TO DESTINATION
00106  0849  A9 19                LDA #$19
00107  084B  85 FD                STA $FD
00108  084D  A9 08                LDA #>SOURCE    ; SET UP POINTER TO CODE SOURCE
00109  084F  85 FC                STA $FC
00110  0851  A9 71                LDA #<SOURCE
00111  0853  85 FB                STA $FB
00112  0855  B1 FB         LOOP2   LDA ($FB),Y     ; MOVE CODE TO DESTINATION
00113  0857  91 FD                STA ($FD),Y
00114  0859  C8                   INY
00115  085A  D0 F9                BNE LOOP2
00116  085C  E6 FE                INC $FE
00117  085E  E6 FC                INC $FC
00118  0860  A5 FC                LDA $FC
00119  0862  C9 0B                CMP #$0B        ; MOVE THREE PAGES TOTAL
00120  0864  90 EF                BCC LOOP2
00121  0866  A2 00                LDX #$00        ; LOAD X,Y WITH POINTER TO -
00122  0868  A0 5C                LDY #$5C        ; NEW TOP OF FREE MEMORY.
00123  086A  18                   CLC             ; JUMP TO KERNAL SUBROUTINE TO -
00124  086B  20 99 FF             JSR $FF99       ; SET MEMORY TOP.
00125  086E  4C 02 E4             JMP $E402       ; TELL BASIC ABOUT MEMORY TOP
00126  0871                ;
00127  0871                SOURCE
00128  0871                ;
00129  0871  88                   .BYTE $88,$1F,$00,$00,$00 ; 159 FLOATING POINT
00129  0872  1F
00129  0873  00
00129  0874  00
00129  0875  00
00130  0876  87                   .BYTE $87,$46,$00,$00,$00 ; 99 FLOATING POINT
00130  0877  46
00130  0878  00
00130  0879  00
00130  087A  00
```

More

Listing continued.

```
00131  087B  83                .BYTE $83,$16,$CB,$E3,$F8 ; (3/2)*PI
00131  087C  16
00131  087D  CB
00131  087E  E3
00131  087F  F8
00132  0880  7B                .BYTE $7B,$0E,$FA,$35,$12 ; PI/180
00132  0881  0E
00132  0882  FA
00132  0883  35
00132  0884  12
00133  0885  80                .BYTE $80,$40,$00,$00,$00 ; .75
00133  0886  40
00133  0887  00
00133  0888  00
00133  0889  00
00134  088A                ;
00135  088A                GR
00136  088A                ;
00137  088A  48                   PHA
00138  088B  20 73 00             JSR CHRGET       ; ADVANCE TEXT POINTER
00139  088E  68                   PLA
00140  088F  C9 30                CMP #'0          ; WAS LAST CHARACTER A ZERO?
00141  0891  F0 0C                BEQ GR0
00142  0893  C9 31                CMP #'1          ; WAS CHARACTER A ONE?
00143  0895  D0 17                BNE EXIT
00144  0897  A9 3B          GR1   LDA #$3B
00145  0899  A0 C6                LDY #$C6
00146  089B  A2 7D                LDX #$7D
00147  089D  D0 06                BNE SCREEN
00148  089F  A9 1B          GR0   LDA #$1B
00149  08A1  A0 C7                LDY #$C7
00150  08A3  A2 15                LDX #$15
00151  08A5  8D 11 D0       SCREEN STA $D011       ; SET/CLEAR BIT-MAP MODE
00152  08A8  8C 00 DD             STY $DD00        ; SELECT VIC BANK (1 FOR GR#1)
00153  08AB  8E 18 D0             STX $D018        ; SELECT SCREEN + COLOR MEMORY
00154  08AE  60             EXIT  RTS
00155  08AF                ;
00156  08AF                COLR
00157  08AF                ;
00158  08AF  20 EB B7             JSR INPUT2       ; INPUT COLORS
00159  08B2  8A                   TXA              ; GET FOREGROUND COLOR
00160  08B3  0A                   ASL A            ; MOVE INTO HIGH NYBBLE OF A
00161  08B4  0A                   ASL A
00162  08B5  0A                   ASL A
00163  08B6  0A                   ASL A
00164  08B7  05 14                ORA $14          ; GET BACKGROUND COLOR
00165  08B9  A2 5C                LDX #$5C
00166  08BB  86 FC                STX $FC          ; SET UP POINTER TO COLOR MEMORY
00167  08BD  A0 00                LDY #$00
00168  08BF  84 FB                STY $FB
00169  08C1  91 FB          FILL  STA ($FB),Y      ; FILL COLOR MEMORY
00170  08C3  C8                   INY
00171  08C4  D0 FB                BNE FILL
00172  08C6  E6 FC                INC $FC
00173  08C8  A6 FC                LDX $FC
00174  08CA  E0 60                CPX #$60         ; $6000 = TOP OF COLOR MEMORY
00175  08CC  90 F3                BCC FILL
00176  08CE  60                   RTS
00177  08CF                ;
00178  08CF                LEFT
00179  08CF                ;
00180  08CF  20 9E AD             JSR EVALEX       ; INPUT ANGLE
00181  08D2  A5 66                LDA $66          ; NEGATE SIGN OF ANGLE -
00182  08D4  49 FF                EOR #$FF         ; TURNS 'LEFT 90'-
00183  08D6  85 66                STA $66          ; INTO 'RIGHT -90'.
00184  08D8  50 03                BVC RIGHT+3
00185  08DA                ;
00186  08DA                RIGHT
00187  08DA                ;
00188  08DA  20 9E AD             JSR EVALEX       ; INPUT ANGLE
00189  08DD                       MULPLY DEGRAD    ; CONVERT TO RADIANS
00196  08E4                       ADD CURANG       ; CURANG=CURANG+NEW ANGLE
00203  08EB                       UPDATE CURANG
00210  08F2  20 64 E2             JSR COSINE
00211  08F5                       UPDATE DX        ; DX=COS(CURANG)
00218  08FC                       FETCH CURANG
00225  0903  20 6B E2             JSR SINE
00226  0906                       MULPLY SQUARE
00233  090D                       XLOYHI DY        ; DY=SIN(CURANG)*SQUARE
00237  0911  4C D7 BB             JMP FACMEM       ; UPDATE DY
00238  0914                ;
00239  0914                PLOT
00240  0914                ;
00241  0914  20 EB B7             JSR INPUT2       ; INPUT COORDINATES
00242  0917  86 02                STX $02          ; STORE YCOR
00243  0919  A9 C7                LDA #$C7         ; YCOR = 199-YCOR: THIS PUTS -
00244  091B  E5 02                SBC $02          ; THE SCREEN ORIGIN IN LOWER -
00245  091D  85 02                STA $02          ; LEFT-HAND CORNER.
00246  091F  A5 15                LDA $15          ; XCOR RETURNED IN $14,$15
```

More →

A=USR(.8).

How To

Given the interesting architecture of the Commodore-64, there are at least three ways new keywords can be added to Basic. There's the tried-and-true wedge, which unfortunately slows down the execution of the Basic interpreter.

A better method is to copy the Basic ROM into the RAM occupying the same address and then modify the keywords and keyword vectors to suit your needs. The problem here is that you lose an existing command every time you add a new one. I started my project using this method, but decided that I didn't want to eliminate the use of DEF FN or VERIFY, to name only two keywords. So I chose another method.

When you type a line of Basic and press return, the Basic interpreter parses the line and converts keywords into tokens. If the interpreter is expecting a keyword but doesn't find one, LET is assumed and the interpreter jumps to the LET routine.

Until now, that is. Changing this jump so that it points to a new parsing routine lets you add new keywords, although they won't be tokenized. This routine doesn't slow the parsing of the standard keywords, except for an assumed LET. But, since the assumed LET travels a tortuous route already (mismatching with 35 keywords), the addition of new keywords is only a slight detour. If you ever find the computer rejecting the turtle graphics commands, the command Poke 1,54 should reenable them.

You need a machine language monitor or an assembler to enter this program. Listing 1 is in assembly language; Listing 2 is a machine language dump. Both are the same program—I've provided two forms for your convenience. When you're through entering the program, save it, starting at hexadecimal address 0801 (rather than 0800 hex). It will then load and run as a Basic program.

The demonstration program (Listing 3) should give you some ideas on using turtle graphics. Notice how circles are drawn. You'll find that, in many cases, just changing one number creates an entirely different picture.

If you're unable or unwilling to enter this program, I'll be happy to make a copy for you. Just send four dollars and a tape or disk along with a stamped, self-addressed mailer.■

Listing continued.

```
00247  0921  F0 0B          BEQ OK          ; IF XCOR IS OFF SCREEN -
00248  0923  C9 01          CMP #$01        ; THEN DO NOT PLOT.
00249  0925  D0 06          BNE NOTOK
00250  0927  A5 14          LDA $14
00251  0929  C9 40          CMP #$40
00252  092B  90 01          BCC OK
00253  092D  60        NOTOK RTS
00254  092E  A5 02      OK    LDA $02         ; IF YCOR IS OFF SCREEN
00255  0930  C9 C8          CMP #$C8        ; THEN DO NOT PLOT.
00256  0932  B0 F9          BCS NOTOK
00257  0934             ; IN ORDER TO SELECT THE CORRECT PIXEL TO SET
00258  0934             ; THE FOLLOWING EQUATIONS ARE USED -
00259  0934             ; CHAR = BASE+40*INT(Y/8)+INT(X/8)
00260  0934             ; BYTE = YAND7, BIT = XAND7
00261  0934  29 07          AND #$07
00262  0936  A8             TAY             ; Y HOLDS BYTE
00263  0937  A9 00          LDA #$00
00264  0939  85 FC          STA $FC
00265  093B  A5 02          LDA $02
00266  093D  29 F8          AND #$F8        ; INT(Y/8)
00267  093F  2A             ROL A
00268  0940  26 FC          ROL $FC
00269  0942  2A             ROL A
00270  0943  26 FC          ROL $FC
00271  0945  2A             ROL A
00272  0946  26 FC          ROL $FC         ; INT(Y/8)*8
00273  0948  85 FB          STA $FB         ; SAVE INTERMEDIATE VALUE
00274  094A  A6 FC          LDX $FC
00275  094C  2A             ROL A
00276  094D  26 FC          ROL $FC
00277  094F  2A             ROL A
00278  0950  26 FC          ROL $FC         ; INT(Y/8)*32
00279  0952  65 FB          ADC $FB         ; ADD INT(Y/8)*8 LOW BYTE
00280  0954  85 FB          STA $FB
00281  0956  8A             TXA
00282  0957  65 FC          ADC $FC         ; ADD INT(Y/8)*8 HIGH BYTE
00283  0959  09 60          ORA #$60        ; ADD BASE ($6000)
00284  095B  AA             TAX
00285  095C  A5 14          LDA $14
00286  095E  29 F8          AND #$F8        ; INT(X/8)
00287  0960  65 FB          ADC $FB         ; ADD LOW BYTE INT(X/8)
00288  0962  85 FB          STA $FB
00289  0964  8A             TXA
00290  0965  65 15          ADC $15         ; ADD HIGH BYTE INT(X/8)
00291  0967  A5 FC          STA $FC
00292  0969  A5 14          LDA $14
00293  096B  29 07          AND #$07
00294  096D  AA             TAX             ; X HOLDS BIT
00295  096E  B1 FB          LDA ($FB),Y     ; SELECT CHAR,BYTE
00296  0970  1D 1F 82       ORA BITVAL,X    ; SET SPECIFIED BIT
00297  0973  91 FB          STA ($FB),Y
00298  0975  60             RTS
00299  0976             ;
00300  0976             DRAW
00301  0976             ;
00302  0976  A9 E6          LDA #$E6        ; #$E6=LT BLUE ON DK BLUE
00303  0978  85 97          STA FLAG        ; SET TAIL DOWN
00304  097A  20 61 80       JSR COLOR
00305  097D  A0 0F          LDY #$0F
00306  097F  B9 18 80   INITXY LDA STARTX-1,Y ; SET UP XCOR, YCOR AND ANGLE
00307  0982  99 09 80       STA XCOR-1,Y
00308  0985  88             DEY
00309  0986  D0 F7          BNE INITXY
00310  0988  84 FB          STY $FB
00311  098A  A9 60          LDA #$60        ; SET UP POINTER TO SCREEN
00312  098C  85 FC          STA $FC
00313  098E  98             TYA
00314  098F  91 FB      CLEAR STA ($FB),Y    ; CLEAR SCREEN
00315  0991  C8             INY
00316  0992  D0 FB          BNE CLEAR
00317  0994  E6 FC          INC $FC
00318  0996  10 F7          BPL CLEAR
00319  0998  20 91 B3       JSR FIXFLO      ; PUT ZERO INTO FACC#1
00320  099B  20 85 80       JSR FINDDX      ; INITIALIZE DX AND DY
00321  099E  4C 3F 80       JMP IGR1        ; TURN ON GRAPHIC SCREEN
00322  09A1             ;
00323  09A1             TAIL
00324  09A1             ;
00325  09A1  E9 55          SBC #'U         ; ANY CHARCTER OTHER THAN 'U' -
00326  09A3  85 97          STA FLAG        ; WILL SET TAIL DOWN.
00327  09A5  20 73 00   IGNORE JSR CHRGET    ; IGNORE ALL BUT FIRST CHAR
00328  09A8  D0 FB          BNE IGNORE
00329  09AA  60             RTS
00330  09AB             ;
00331  09AB             MOVE
00332  09AB             ;
00333  09AB  20 9E AD       JSR EVALEX      ; INPUT LENGTH OF MOVE
00334  09AE  A5 66          LDA $66
00335  09B0  30 69          BMI DONE        ; IF NEGATIVE LENGTH RETURN
00336  09B2  20,9B BC       JSR FLOFIX
```

More

Listing continued.

```
00337  09B5  A6 65          LDX $65
00338  09B7  A5 64          LDA $64
00339  09B9  F0 03          BEQ SHORT       ; IF 'MOVE 511' X HOLDS #$FF -
00340  09BB  A9 FF          LDA #$FF        ; BUT GETS INCREMENTED TO #$00 -
00341  09BD  E8             INX             ; HENCE THE UNPREDICTABILITY.
00342  09BE  85 19    SHORT STA $19
00343  09C0  86 1A          STX $1A
00344  09C2           NEXT  FETCH XCOR
00351  09C9                 ADD DX          ; XCOR = XCOR + DX
00358  09D0                 UPDATE XCOR
00365  09D7  20 9B BC       JSR FLOFIX
00366  09DA  A6 65          LDX $65
00367  09DC  A4 64          LDY $64
00368  09DE  84 15          STY $15         ; STORE FOR USE BY PLOT
00369  09E0  86 14          STX $14
00370  09E2                 FETCH YCOR
00377  09E9                 ADD DY          ; YCOR = YCOR + DY
00384  09F0                 UPDATE YCOR
00391  09F7  20 9B BC       JSR FLOFIX
00392  09FA  A6 64          LDX $64         ; ELIMINATE ANY YCOR > 255
00393  09FC  D0 0B          BNE NEXT1
00394  09FE  A6 65          LDX $65
00395  0A00  86 02          STX $02
00396  0A02  A5 97          LDA FLAG        ; IF TAIL IS UP DO NOT PLOT
00397  0A04  F0 03          BEQ NEXT1
00398  0A06  20 C7 80       JSR PLOT1
00399  0A09  A5 19    NEXT1 LDA $19
00400  0A0B  F0 05          BEQ NEXT2
00401  0A0D  C6 19          DEC $19
00402  0A0F  18             CLC
00403  0A10  90 B0    STEP  BCC NEXT
00404  0A12  A5 1A    NEXT2 LDA $1A
00405  0A14  F0 05          BEQ DONE
00406  0A16  C6 1A          DEC $1A
00407  0A18  18             CLC
00408  0A19  90 F5          BCC STEP
00409  0A1B  60      DONE  RTS
00410  0A1C                 ;
00411  0A1C           ERROR
00412  0A1C                 ;
00413  0A1C  8A             TXA
00414  0A1D  30 08          BMI NOERR
00415  0A1F  48             PHA             ; RETURN TO TEXT SCREEN -
00416  0A20  20 47 80       JSR IGR0        ; TO DISPLAY ERROR MESSAGE.
00417  0A23  68             PLA
00418  0A24  4C 3B A4       JMP IERROR
00419  0A27  4C 74 A4 NOERR JMP READY
00420  0A2A                 ;
00421  0A2A           USR
00422  0A2A                 ;
00423  0A2A                 UPDATE SQUARE   ; NEW SQUARE VALUE
00430  0A31  4C A4 80       JMP FINDDY      ; ADJUST DY
00431  0A34                 ;
00432  0A34           PARSER
00433  0A34                 ;
00434  0A34  A2 00          LDX #$00
00435  0A36  A0 00          LDY #$00
00436  0A38  B1 7A    SEARCH LDA ($7A),Y
00437  0A3A  DD 27 82       CMP TABLE,X     ; MATCH FIRST LETTER OF WORD -
00438  0A3D  F0 0C          BEQ MATCH       ; WITH WORD IN TABLE.
00439  0A3F  8A             TXA
00440  0A40  18       MISSED CLC            ; NO MATCH -
00441  0A41  69 08          ADC #$08        ; POINT X TO NEXT ENTRY
00442  0A43  AA             TAX
00443  0A44  C9 40          CMP #$40        ; AT END OF TABLE?
00444  0A46  D0 F0          BNE SEARCH
00445  0A48  4C A5 A9       JMP LET         ; IF YES JUMP TO LET ROUTINE
00446  0A4B  C8       MATCH INY
00447  0A4C  E8             INX
00448  0A4D  B1 7A          LDA ($7A),Y     ; MATCH ENTIRE WORD WITH TABLE
00449  0A4F  DD 27 82       CMP TABLE,X
00450  0A52  F0 F7          BEQ MATCH
00451  0A54  09 80          ORA #$80        ; LAST CHAR OF WORDS IN TABLE -
00452  0A56  DD 27 82       CMP TABLE,X     ; HAVE HIGH BIT SET.
00453  0A59  F0 07          BEQ FOUND
00454  0A5B  8A             TXA             ; NO MATCH
00455  0A5C  29 F8          AND #$F8
00456  0A5E  A0 00          LDY #$00
00457  0A60  F0 DE          BEQ MISSED      ; GO BACK AND TRY AGAIN
00458  0A62  BD 28 82 FOUND LDA TABLE+1,X   ; PUSH VECTOR ONTO STACK
00459  0A65  48             PHA
00460  0A66  BD 29 82       LDA TABLE+2,X
00461  0A69  48             PHA
00462  0A6A  8A             TXA
00463  0A6B  29 07          AND #$07        ; GET LENGTH OF WORD
00464  0A6D  AA             TAX
00465  0A6E  20 73 00 MOVEP JSR CHRGET      ; ADVANCE TEXT POINTER -
00466  0A71  CA             DEX             ; TO END OF WORD.
00467  0A72  D0 FA          BNE MOVEP
```

More →

Listing continued.

```
00468  0A74  4C 73 00          JMP CHRGET      ; JUMP TO ROUTINE VIA CHRGET
00469  0A77                ;
00470  0A77                ;BITVAL
00471  0A77                ;
00472  0A77  80                    .BYTE $80,$40,$20,$10,$08,$04,$02,$01
00472  0A78  40
00472  0A79  20
00472  0A7A  10
00472  0A7B  08
00472  0A7C  04
00472  0A7D  02
00472  0A7E  01
00473  0A7F                ;
00474  0A7F                ;TABLE OF KEWORDS AND VECTORS
00475  0A7F                ;
00476  0A7F  54 41 49              .BYTE 'TAI',$CC,$81,$48,$00,$00
00476  0A82  CC
00476  0A83  81
00476  0A84  48
00476  0A85  00
00476  0A86  00
00477  0A87  52 49                 .BYTE 'RIGH',$D4,$80,$81,$00
00477  0A8B  D4
00477  0A8C  80
00477  0A8D  81
00477  0A8E  00
00478  0A8F  50 4C 4F              .BYTE 'PLO',$D4,$80,$BB,$00,$00
00478  0A92  D4
00478  0A93  80
00478  0A94  BB
00478  0A95  00
00478  0A96  00
00479  0A97  4D 4F 56              .BYTE 'MOV',$C5,$81,$52,$00,$00
00479  0A9A  C5
00479  0A9B  81
00479  0A9C  52
00479  0A9D  00
00479  0A9E  00
00480  0A9F  4C 45 46              .BYTE 'LEF',$D4,$80,$76,$00,$00
00480  0AA2  D4
00480  0AA3  80
00480  0AA4  76
00480  0AA5  00
00480  0AA6  00
00481  0AA7  47 52                 .BYTE 'GR',$A3,$80,$31,$00,$00,$00
00481  0AA9  A3
00481  0AAA  80
00481  0AAB  31
00481  0AAC  00
00481  0AAD  00
00481  0AAE  00
00482  0AAF  44 52 41              .BYTE 'DRA',$D7,$81,$1D,$00,$00
00482  0AB2  D7
00482  0AB3  81
00482  0AB4  1D
00482  0AB5  00
00482  0AB6  00
00483  0AB7  43 4F 4C              .BYTE 'COL',$D2,$80,$56,$00,$00
00483  0ABA  D2
00483  0ABB  80
00483  0ABC  56
00483  0ABD  00
00483  0ABE  00
00484  0ABF                       .END
SYMBOL TABLE
SYMBOL VALUE
 ADD      FFFF    ALOYHI   FFFF    BASIC    0801    BITVAL   821F
 CHRGET   0073    CLEAR    098F    COLOR    8061    COLR     08AF
 COSINE   E264    CURANG   8014    DEGRAD   8028    DONE     0A1B
 DRAW     0976    DX       8000    DY       8005    ERROR    0A1C
 EVALEX   AD9E    EXIT     08AE    FACMEM   BBD7    FETCH    FFFF
 FILL     08C1    FINDDX   8085    FINDDY   80A4    FIXFLO   B391
 FLAG     0097    FLOFIX   BC9B    FOUND    0A62    GR       088A
 GR0      089F    GR1      0897    IERROR   A43B    IGNORE   09A5
 IGR0     8047    IGR1     803F    INIT     080E    INITXY   097F
 INPUT2   B7EB    LEFT     08CF    LET      A9A5    LOOP1    081A
 LOOP2    0855    MATCH    0A4B    MEMFAC   BBA2    MISSED   0A40
 MOVE     09AB    MOVEP    0A6E    MULPLY   FFFF    MULT     BA28
 NEXT     09C2    NEXT1    0A09    NEXT2    0A12    NOERR    0A27
 NOTOK    092D    OK       092E    PARSER   0A34    PLOT     0914
 PLOT1    80C7    PLUS     B867    READY    A474    RIGHT    08DA
 SCREEN   08A5    SEARCH   0A38    SHORT    09BE    SINE     E26B
 SOURCE   0871    SQUARE   802D    STARTA   8023    STARTX   8019
 STARTY   801E    STEP     0A10    TABLE    8227    TAIL     09A1
 UPDATE   FFFF    USR      0A2A    XCOR     800A    XLOYHI   FFFF
 YCOR     800F

END OF ASSEMBLY
```

Listing 2. Machine language dump of the turtle graphics routines.

```
B*
   PC   SR AHILOADWORK
.;0800 30 00 00 00 F6
.:0800 00 0C 08 0A 00 9E 20 32
.:0808 30 36 32 00 00 00 A9 37
.:0810 85 01 A9 A0 85 FC A0 00
.:0818 84 FB B1 FB 91 FB C8 D0
.:0820 F9 E6 FC 24 FC 50 F3 A9
.:0828 36 85 01 A9 81 8D 01 03
.:0830 8D 12 03 8D 06 A8 A9 C4
.:0838 8D 00 03 A9 D2 8D 11 03
.:0840 A9 DC 8D 05 A8 A9 80 85
.:0848 FE A9 19 85 FD A9 08 85
.:0850 FC A9 71 85 FB B1 FB 91
.:0858 FD C8 D0 F9 E6 FE E6 FC
.:0860 A5 FC C9 0B 90 EF A2 00
.:0868 A0 5C 16 20 99 FF 4C 02
.:0870 E4 88 1F 00 00 00 87 46
.:0878 00 00 00 83 16 CB E3 F8
.:0880 7B 0E FA 35 12 80 40 00
.:0888 00 00 48 20 73 00 68 C9
.:0890 30 F0 0C C9 31 D0 17 A9
.:0898 3B A0 C6 A2 7D D0 06 A9
.:08A0 1B A0 C7 A2 15 8D 11 D0
.:08A8 8C 00 DD 8E 18 D0 60 20
.:08B0 EB B7 8A 0A 0A 0A 0A 05
.:08B8 14 A2 5C 86 FC A0 00 84
.:08C0 FB 91 FB C8 D0 FB E6 FC
.:08C8 A6 FC E0 60 90 F3 60 20
.:08D0 9E AD A5 66 49 FF 85 66
.:08D8 50 03 20 9E AD A9 28 A0
.:08E0 80 20 28 BA A9 14 A0 80
.:08E8 20 67 B8 A2 14 A0 80 20
.:08F0 D7 BB 20 64 E2 A2 00 A0
.:08F8 80 20 D7 BB A9 14 A0 80
.:0900 20 A2 BB 20 6B E2 A9 2D
.:0908 A0 80 20 28 BA A2 05 A0
.:0910 80 4C D7 BB 20 EB B7 86
.:0918 02 A9 C7 E5 02 85 02 A5
.:0920 15 F0 0B C9 01 D0 06 A5
.:0928 14 C9 40 90 01 60 A5 02
.:0930 C9 C8 B0 F9 29 07 A8 A9
.:0938 00 85 FC A5 02 29 F8 2A
.:0940 26 FC 2A 26 FC 2A 26 FC
.:0948 85 FB A6 FC 2A 26 FC 2A
.:0950 26 FC 65 FB 85 FB 8A 65
.:0958 FC 09 60 AA A5 14 29 F8
.:0960 65 FB 85 FB 8A 65 15 85
.:0968 FC A5 14 29 07 AA B1 FB
.:0970 1D 1F 82 91 FB 60 A9 E6
.:0978 85 97 20 61 80 A0 0F B9
.
.:0980 16 80 99 09 80 88 D0 F7
.:0988 84 FB A9 60 85 FC 98 91
.:0990 FB C8 D0 FB E6 FC 10 F7
```

More

Listing continued.

```
.:0998 20 91 B3 20 85 80 4C 3F
.:09A0 80 E9 55 85 97 20 73 00
.:09A8 D0 FB 60 20 9E AD A5 66
.:09B0 30 69 20 9B BC A6 65 A5
.:09B8 64 F0 03 A9 FF E8 85 19
.:09C0 86 1A A9 0A A0 80 20 A2
.:09C8 BB A9 00 A0 80 20 67 B8
.:09D0 A2 0A A0 80 20 D7 BB 20
.:09D8 9B BC A6 65 A4 64 84 15
.:09E0 86 14 A9 0F A0 80 20 A2
.:09E8 BB A9 05 A0 80 20 67 B8
.:09F0 A2 0F A0 80 20 D7 BB 20
.:09F8 9B BC A6 64 D0 0B A6 65
.:0A00 86 02 A5 97 F0 03 20 C7
.:0A08 80 A5 19 F0 05 C6 19 18
.:0A10 90 B0 A5 1A F0 05 C6 1A
.:0A18 18 90 F5 60 8A 30 08 48
.:0A20 20 47 80 68 4C 3B A4 4C
.:0A28 74 A4 A2 2D A0 80 20 D7
.:0A30 BB 4C A4 80 A2 00 A0 00
.:0A38 B1 7A DD 27 82 F0 0C 8A
.:0A40 18 69 08 AA C9 40 D0 F0
.:0A48 4C A5 A9 C8 E8 B1 7A DD
.:0A50 27 82 F0 F7 09 80 DD 27
.:0A58 82 F0 07 8A 29 F8 A0 00
.:0A60 F0 DE BD 28 82 48 BD 29
.:0A68 82 48 8A 29 07 AA 20 73
.:0A70 00 CA D0 FA 4C 73 00 80
.:0A78 40 20 10 08 04 02 01 54
.:0A80 41 49 CC 81 48 00 00 52
.:0A88 49 47 48 D4 80 81 00 50
.:0A90 4C 4F D4 80 BB 00 00 4D
.:0A98 4F 56 C5 81 52 00 00 4C
.:0AA0 45 46 D4 80 76 00 00 47
.:0AA8 52 A3 80 31 00 00 00 44
.:0AB0 52 41 D7 81 1D 00 00 43
.:0AB8 4F 4C D2 80 56 00 00 DF
.
```

```
1 REM TURTLE DEMO
10 DRAW:COLR14,6:TAILUP:RIGHT150:MOVE150:LEFT150:TAILDOWN
20 FORG=1TO36:FORT=1TO2:FORR=1TO8:MOVE9:LEFT10:NEXT
30 FORR=1TO6:MOVE9:RIGHT10:NEXTR,T:RIGHT170:NEXT:GOSUB300
40 FORT=4TO8STEP2:A=360/T:DRAW
50 FORR=1TOT:FORG=1TOT:MOVEA:LEFTA:NEXT:RIGHTA:NEXT
60 GOSUB300:NEXT
70 DRAW:POKE53280,0:COLR0,2:A=160:FORT=1TO200:MOVET:LEFTA:NEXT:GOSUB300
80 FORT=13TO15STEP2:A=720/T:DRAW
90 FORR=1TOT:FORG=1TOT:TAILDOWN:IFG/2=INT(G/2)THENTAILUP
100 MOVEA:LEFTA:NEXT:RIGHTA:NEXT
110 GOSUB300:NEXT
120 DRAW:POKE53280,0:COLR0,6:FORT=1TO24:FORR=1TO36
130 MOVE10:LEFT10:NEXT:RIGHT15:NEXT:GOSUB300
140 DRAW:COLR6,8:A=89.5:FORT=1TO230STEP1.5:MOVET:LEFTA:NEXT:GOSUB300
299 GR#0:END
300 FORY=1TO1000:NEXT:POKE53280,14:RETURN
```

Listing 3. Demonstration program.

ANNOUNCING
the premiere of

jr®

Learning to use your *IBM PCjr* becomes easy and fun with **jr** magazine. Because **jr** keeps things simple, the whole family will *learn and understand while they enjoy themselves.*

jr will explain how the *PCjr* works and guide you step-by-step in setting up your system. Most of all **jr** will answer your most important questions month after month . . . like

- **What can I use my *PCjr* for? jr** magazine will teach you how to prepare household budgets, balance your checkbooks, do your taxes and organize schedules and lists.
- **What about the kids?** From kindergarten to college, **jr** helps the kids with educational programs in every subject area.
- **Are there more serious uses for *PCjr*?** Yes! You'll improve your own business skills by learning about word processing, spreadsheets and graphics.
- **What about having fun with *PCjr*?** Relax and enjoy all the latest games with your whole family.

Plus:

- **What programs should I choose for the *PCjr*? jr** magazine will tell you what programs on the market will run on the *PCjr*, and which ones won't. Plus what new programs are on the way. **jr** will evaluate them *before* you buy.

Order your subscription to **jr** today and you'll get the special charter subscription rate of $14.97. That's 25% off the basic subscription rate! Plus, with your pre-payment of check, American Express, Visa or Mastercard we'll send you a FREE issue making a total of 13 issues for $14.97.

Take advantage of this incredible charter subscription offer. Mail the attached card, the coupon or call TOLL FREE 1-800-258-5473. In New Hampshire call 1-924-9471 . . . TODAY!

Sign me up as a charter subscriber to **jr.** I understand that with prepayment of check or credit card I'll receive a FREE issue making a total of 13 issues for $14.97. Save $5.00 off the basic rate!

☐ CHECK/MO ☐ MC ☐ AE ☐ VISA ☐ BILL ME

Card # _____ Exp. Date _____

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____

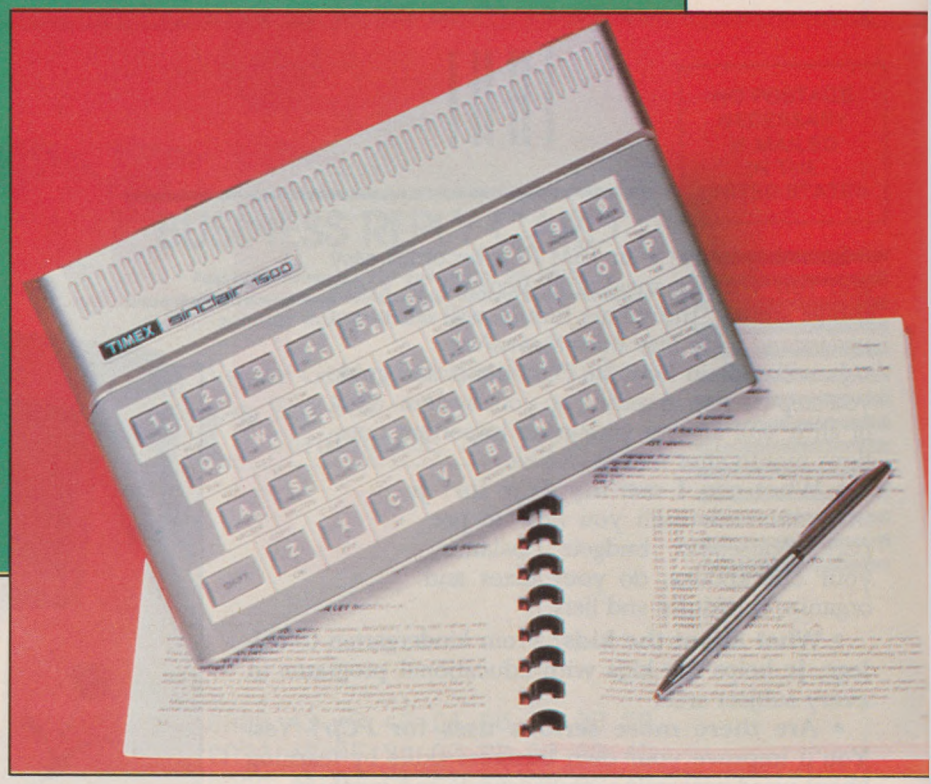
Canada & Mexico $17.97, 1 year only, US funds drawn on US bank.
Foreign Surface $34.97, 1 year only, US funds drawn on US bank.
Foreign airmail, please inquire. Please allow 6-8 weeks for delivery.

343F7

jr® For IBM's Home Computer
Box 903 Farmingdale, NY 11737

T/S: Is 1500 > 1000?

With a keyboard and 16Kb RAM, the T/S 1500 is greater than its older and less powerful brother, the T/S 1000. But the question is, "Is there a market for the 1500?"



By Sharon Zardetto Aker

Calling the Timex/Sinclair 1500 "just a 1000 with a keyboard and built-in 16Kb RAM" would be unfair. The use of the word "just" underestimates both the excellence of the original computer and the importance of the two changes.

The T/S 1000, despite its reputation, was a marvelous little machine whose size belied its power. The American version of Clive Sinclair's phenomenally popular ZX81 quickly gained a faithful group of users who had only two complaints: the keyboard was terrible, and the add-on RAM module kept falling off.

Timex addresses those problems in its T/S 1500. (If you want color, sound, hi-res graphics and more, Timex has come up with a whole new computer, the T/S 2068 color computer.)

Is there any place in today's home computer market for a no-color, no-sound, lo-res graphics machine? Or, perhaps of more interest to Timex, is there a place in the school market for such a computer? There should be, because the T/S 1500 is the perfect literacy machine.

The Outside

The rubber chiclet keyboard is an improvement over the 1000's membrane; even judged on its own merit, it isn't bad as chiclets go. Although you have to catch the key dead-center for it to register, you can hear and feel a slight pop when you hit it correctly.

The 40 keys are imprinted with white letters, numbers and graphics symbols, with shifted symbols in black. There is no space bar, and although the space key is a little larger than the others, it is placed in the lower corner where you would normally expect to find a shift key.

Contrary to prerelease rumors, the space key still doubles as a break key, as it does on the 1000. There is only one shift key, on the opposite side of the keyboard from the space key. So even though the keys are comfortably spaced, touch typing is not an easy project.

All four jacks—TV, power, ear and microphone—have been moved to the back of the case. This makes the recorder connections more awkward: since you can't leave both ear and mic connections in when saving or loading programs, there is a lot of plugging and unplugging to be done in the course of a computing session. Unless you lift the computer and stand it on its edge, you will have to resort to some tricky contortions to see what you're doing. At least the cassette connections are on the opposite edge

Address correspondence to Sharon Zardetto Aker, 20 Courtland Drive, Sussex, NJ 07461.

from the power plug; I have too often pulled the wrong wire out after loading a program into my 1000.

Step to the Rear

The peripheral port (Sinclair standard, not any industry standard) is still located at the rear, but it affords a sturdier connection for the RAM pack than was available on the 1000.

At 1¼ pounds, the 1500 is nearly twice the weight of the 1000; it has just enough heft to make you think there is really something inside (that was always hard to believe about the 1000), and it sits firmly on a table without sliding around.

The Inside

The inside of the 1500, with the exception of additional RAM, is identical to the 1000. The Z80A microprocessor runs at 3.5 MHz. There is 8Kb of ROM, and its 16Kb RAM can be doubled by using the RAM pack designed for the 1000.

There is a built-in rf modulator; with only one video output, you're restricted to a television-only display.

Loads of Problems?

The 1500 is supposed to work with any cassette recorder. The 1000 was finicky about just what recorder it would work with, and I have no doubt that the 1500 is the same.

Loading with the same recorder that was used to save a program isn't a problem, once you find the correct volumn and tone levels to use.

However, when a program is saved on one recorder and loaded from another (which, of course, is the case with all commercial software), there are often compatibility problems. Timex is offering two solutions: a list of recorders that have proven to be widely compatible, and Timex's own cassette player.

The second unavoidable problem is the speed (and I use the term loosely) of the save and load operations. A full five minutes for loading a long program isn't unusual. Whether or not the new T/S drive that is under development will be compatible with the 1500 remains to be seen.

No-Display Mode

The T/S can work in either Fast or Slow mode. In Slow mode, calculations are done between each screen refresh, a method of operation that slows the running of the program. The fast mode, as you would expect, runs significantly faster; however, there is no display in that mode except for the final screen when the program ends.

Sinclair has unintentionally encouraged hundreds of people to learn machine code programming, because only with machine code can you keep the display on the screen and keep the program moving along, too.

When the display is on the screen, it is 24 lines of 32 columns; you can use only 22 of the lines, because Sinclair computers use the bottom portion of the screen for input information and prompts. Actually, a 24-line display for a program that needs no input is easy to achieve with a well-placed poke.

The character set consists of numerals, uppercase letters, the usual mathematical symbols and most punctuation—the apostrophe and exclamation being notably absent. It includes the unusual "quote image," which allows you to print quotation marks inside strings without resorting to character codes. The graphics set consists of 20 boxes whose quarters are variously shaded white, gray and black. All characters are available in both true and inverse video.

Editing a T/S program is a lesson in patience—or perhaps in frustration. There is no full-screen editing.

Sinclair Basic

The major features of Sinclair Basic, both particularly appropriate for beginners, are its one-key command entry and the built-in syntax editor.

Each of the 1500's keys can represent up to five things: a keyword, a function, a letter or number, a shifted symbol and a graphics character. The D key, for instance, besides its letter, is also used for the DIM statement, the Slow command, the Arctangent function and a graphics square. This is not a confusing system to work with—the on-screen cursor, together with shifting or double-shifting, determines how a press of the key will be interpreted.

A version of Sinclair's syntax editor should be incorporated into every computer: no one would ever have a program stop because of a misplaced semicolon!

In Sinclair Basic, simple numeric variables can be of any length, another plus for neophytes, because SCORE can stand for the score and PLAYER for the player. String and array variables are limited to a single letter; string array storage is nothing short of Procrustean.

There is a full range of mathmatical and trigonometric functions, with numbers stored to an accuracy of 9½ decimal places.

All the usual string functions are also available, and although the slicing techniques aren't standard ones, I find them easier to use.

In most cases, there are no parentheses needed around arguments. This feature, along with the automatic spacing before and after Basic words, makes for a clean, easy-to-read program listing.

Practice Your Patience

Editing a T/S program is a lesson in patience—or perhaps in frustration. There is no full-screen editing. A program line must be brought to the work area at the bottom of the screen to be edited. The vertical cursor controls can only be used to move what is called the program cursor, which points to the line to be brought down by the Edit command.

Since you can only move your cursor horizontally and there is no repeat function on any key, editing a long line is tedious. Adding to the clumsiness of this editing system is the inconvenience caused by the cursor controls and delete key being shifted functions.

Of course, since the 1500 is aimed at beginners, program lines won't be very long (multiple commands are not available), and you won't have already been spoiled by easier editing systems on other computers.

Another plus for the newcomer is the 1500's excellent documentation. The manual for the 1000 is one of the better ones around; it has been rewritten for the 1500 and is now one of the best.

Graphics?

Graphics is another term to be used

loosely with the Sinclair. Besides designing pictures, moving or still, with the graphics squares, there is also what is rather generously termed medium-resolution graphics available with the Plot command. Plotting on the T/S addresses squares of 4×4 pixels, with coordinates ranging from 0,0 in the lower left corner to 64,44 in the upper right. Using the Unplot command, which erases the line, interesting effects can be achieved with minimal effort. Of course, you won't see anything but the final display if you're running the computer in the Fast mode.

Peripherals

The Timex printer is an excellent piece of equipment and it sells for less than $100. This quick and quiet dot-matrix printer prints 32 cps on inexpensive 4½-inch wide thermal paper, perfect for program listings.

As mentioned before, the 16Kb RAM module designed for the 1000 can also be used with the 1500.

Timex has announced a cartridge-player-attachment that will enable cartridge programs to be used with the 1500; I wouldn't bet on the appearance of either the attachment or the cartridges. The strong point of the 1500, even (or especially) in the educational market is not the programs you can buy to run on it, but the ease with which you can write a program to run.

This is not to imply that some sophisticated programs can't be made for the 1500; I've seen some impressive ones designed for the 1000. However, few, if any, software companies will be writing for the 1500 (most are undecided on support of the T/S 2068, a much more advanced machine) and Timex may not follow through on software support further than an initial line of educational and short game programs.

Since all 1000-compatible peripherals will work with the 1500, expansion possibilities are practically limitless; there are many third-party hardware items available. You can add color, sound, a programmable character generator, a voice box, and 64Kb of memory or even interface a disk drive. Of course, by the time you're finished, you will have spent a lot more than if you had bought a computer with those capabilities in the first place.

Taking Aim

It seems likely that Timex will want to avoid direct competition with its own 2068 color computer. They may not push the 1500 much in the retail market—the lucrative school market is the more likely target.

The Timex literacy machine should be welcomed there with wide-open arms—if there are any wide-open eyes that haven't been blinded by Apple, Commodore and Tandy. As a learning tool, the 1500 puts them all to shame, and at a fraction of the cost.

The T/S 1500 is an excellent machine, at a reasonable price. It might be hard to justify a purchase, however, considering that for about twice the price (and still under $200) you can get a computer with color, sound, hi-res graphics and a better chance of software support. It would be excellent as a second computer if your first one is strictly a software-running machine and you want a painless introduction to programming, or if you want your children to write some games instead of only playing them.■

Circle 169 on Reader Service card.

HOFACKER

Books + Software for ATARI VIC-20 OSI SINCLAIR TIMEX

Payment: check, money order, VISA, MASTER-CHARGE, Euroscheck.
Orders from outside USA: add 15% shipping. CA residents add 6.5% tax
*ATARI is a registered trademark of ATARI Inc.
*VIC-20 is a registered trademark of Commodore

ELCOMP PUBLISHING, INC
53 Redrock Lane
Pomona, CA 91766
Phone: (714) 623 8314

PRODUCTS FOR ATARI* 400/800 FROM ELCOMP

BOOKS:

ATARI BASIC – Learning by Using
An excellent book for the beginner. Many short programs and learning exercises. All important features of the ATARI computers are described (screen drawings, special sounds, keys, paddles, joysticks, specialized screen routines, graphics, sound applications, peeks, pokes, and special stuff). Also suggestions are made that challenge you to change and write program routines.
Order #164 **$7.95**

Games for the ATARI Computer
This book describes advanced programming techniques like player-missile-graphics and use of the hardware-registers. Contains many ready to run programs in BASIC and one called GUNFIGHT in machine language.
Order #162 **$7.95**

Programming in 6502 Machine Language on your PET+CBM
2 complete Editor/Assemblers (Source code 3 hexdump + description plus a powerful machine language monitor (Hexdump)).
Order #166 **$19.95**

How to program your ATARI in 6502 machine language
Introduction to machine language for the BASIC programmer
Order #169 **$9.95**

SOFTWARE IN BASIC FOR ATARI

Invoice Writing for Small Business
This program makes writing invoices easy. Store your products in DATA statements with order-number, description, and price. The program later retrives the description and price matching to the entered order-number. The shipping cost and the discount may be calculated automatically depending on the quantity ordered or entered manually. The description to the program tells you how to change the program and adapt it to your own needs. Comes with a couple of invoice forms to write your first invoices on to it.
Order #7201 **cassette version** **$29.95**
Order #7200 **disk version** **$39.95**

Mailing List
This menu driven program allows the small business man to keep track of vendors and customers. You can search for a name or address of a certain town or for an address with a certain note. 50 addresses are put into one file.
Order #7212 **cassette version** **$19.95**
Order #7213 **disk version** **$24.95**

Inventory Control
This program is menu driven. It gives you the following options: read/store data, define items, entry editing, inventory maintenance (incoming-outgoing), reports. The products are stored with inventory number, manufacturer, reorder level, present level, code number, description.
Order #7214 **cassette version** **$19.95**
Order #7215 **disk version** **$24.95**

Programs from Book #164
The programs from book no. 164 on cassette. (Book included)
Order #7100 **$29.00**

Game Package
Games on cassette. (Bomber, tennis, smart, cannon fodder, etc.)
Order #7216 **$9.95**

Microcomputer Hardware Handbook (845 pages)
Descriptions, pinouts and specifications of the most popular microprocessors and support chips.
A MUST for the hardware buff.
Order-No. 29
$14.95

Care and Feeding of the Commodore PET
Eight chapters exploring PET hardware. Includes repair and interfacing information. Programming tricks and schematics.
Order #150 **$9.95**

SOFTWARE IN MACHINE LANGUAGE for ATARI

ATMONA-1
This is a machine language monitor that provides you with the most important commands for programming in machine-language. Disassemble, dump (hex and ASCII), change memory location, block transfer, fill memory block, save and load machine-language programs, start programs. Printer option via three different interfaces.
Order #7022 **cassette version** **$19.95**
Order #7023 **disk version** **$24.95**
Order #7024 **cartridge version** **$59,00**

ATMONA-2
This is a tracer (debugger) that lets you explore the ATARI RAM/ROM area. You can stop at previously selected address, opcode, or operand. Also very valuable in understanding the microprocessor. At each stop, all registers of the CPU may be changed. Includes ATMONA-1.
Order #7049 **cassette version** **$49.95**
Order #7050 **disk version** **$54.00**

ATMAS
Macro-Assembler for ATARI-800/48k. One of the most powerful editor assemblers on the market. Versatile editor with scrolling. Up to 17k of source-Code. Very fast, translates 5k source-code in about 5 seconds. Source code can be saved on disk or cassette. (Includes ATMONA-1)
Order #7099 **disk version** **$89.00**
Order #7999 **cartridge version** **$129.00**

ATAS
Same as ATMAS but without macro-capability. Cassette-based.
Order #7098 **32k RAM** **$49.95**
Order #7998 **48k RAM** **$49.95**

ATEXT-1
This wordprocessor is an excellent buy for your money. It features screen oriented editing, scrolling, string search (even nested), left and right margin justification. Over 30 commands. Text can be saved on disk or cassette.
Order #7210 **cassette version** **$29.95**
Order #7216 **disk version** **$34.95**
Order #7217 **cartridge version** **$69.00**

GUNFIGHT
This game (8k machine-language) needs two joystieks. Animation and sound. Two cowboys fight against each other. Comes on a bootable cassette.
Order #7207 **$19.95**

FORTH for the ATARI

FORTH from Elcomp Publishing, Inc. is an extended Fig-Forth-version, Editor and I/O package included. Utility package includes decompiler, sector copy, Hex-dump (ASCII), ATARI Filehandling, total graphic and sound, joystick program and player missile. Extremely powerful!
Order #7055 **disk** **$39.95**

Floating point package with trigonometric functions ($0 - 90^0$).
Order #7230 **disk** **$29.95**

Learn-FORTH from Elcomp Publishing, Inc.
A subset of Fig-Forth for the beginner. On disk (32k RAM) or on cassette (16k RAM).
Order #7053 **$19.95**

Expansion boards for the APPLE II

The Custom Apple + Other Mysteries
A complete guide to customizing the Apple Software und Hardware
Order-No. 680 **$24.95**

We also stock the boards which are used in the book "The Custom Apple ..." (barebords)
6522 I/O Board No. 605 **$39.00**
EPROM Burner No. 607 **$49.00**
8K EPROM/RAM Board No. 609 **$29.00**
Prototyping board for the Apple II No. 604 **$29.00**
Slot repeater board for the Apple II No. 606 **$49.00**
Order two boards and get the book free !

COMING SOON ! ORDER NOW !
A Look in the future with your ATARI
(Astrology and how to do your own horoscope on the ATARI 800. **Order No. 171** **$9.95**
FORTH on the ATARI – Learning by Using
Order No. 170 **$7.95**

Hardware – ADD–ONS for ATARI

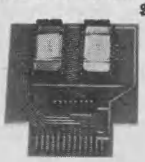
PRINTER INTERFACE
This construction article comes with printed circuit board and software. You can use the EPSON printer without the ATARI printer interface. (Works with gameports 3 and 4).
Order #7211 **$19.95**

RS-232 Interface for your ATARI 400/800
Software with connector and construction article.
Order #7291 **$19.95**

EPROM BURNER for ATARI 400/800
Works with gameports. No additional power supply needed. Comes compl. assembled with software (2716, 2732, 2532).
Order #7042 **$179.00**

EPROM BURNER for ATARI 400/800 KIT
Printed circuit board incl. Software and extensive construction article.
Order #7292 **$49.00**

EPROM BOARD (CARTRIDGE)
Holds two 4k EPROMs (2532). EPROMs not included.
Order #7043 **$29.95**



EPROM BOARD KIT
Same as above but bare board only with description.
Order #7224 **$14.95**

ATARI, VIC-20, Sinclair, Timex and OSI

New – for your ATARI 400/800
Astrology and Biorythm for ATARI (cass. or disk). Order #7223 $29.95
Birth control with the ATARI (Knaus Ogino) Order #7222 cass. or disk $29.95

Books + Software for VIC-20 (requires 3k RAM Exp.)

#4870	Wordprocessor for VIC-20, 8k RAM	$19.95
#4883	Mailing List for VIC-20, 16k RAM	$14.95
#141	Tricks for VICs - The VICstory Progr.	$9.95
#4880	TIC TAC VIC	$9.95
#4881	GAMEPACK I (3 Games)	$14.95
#4885	Dual Joystick Instruction	$9.95

INPUT/OUTPUT Programming with your VIC Order #4886 $9.95

#4896	Miniassembler for VIC-20	$19.95
#4881	Tennis, Squash, Break	$9.95
#4894	Runfill for VIC	$9.95

Universal Experimenter Board for the VIC-20
(Save money with this great board). This board plugs right into the expansion slot of the VIC-20. The board contains a large prototyping area for your own circuit design and expansion. The construction article shows you how to build your own 3k RAM expander and ROM-board.
Order #4844 **$18.95**

Software for SINCLAIR ZX-81 and TIMEX 1000

#2399	Machine Language Monitor	$9.95
#2398	Mailing List	$19.95

Programming in BASIC and machine language with the ZX-81 (82) or TIMEX 1000.
Order #140 (book) $9.95

Books for OSI

#157	The First Book of Ohio	$7.95
#158	The Second Book of Ohio	$7.95
#159	The Third Book of Ohio	$7.95
#160	The Fourth Book of Ohio	$7.95
#161	The Fifth Book of Ohio	$7.95
#151	8K Microsoft BASIC Ref. Man.	$9.95
#152	Expansion Handbook for 6502 and 6802	$9.95
#153	Microcomputer Appl. Notes	$9.95

Complex Sound Generation
New revised applications manual for the Texas Instruments SN 76477 Complex Sound Generator.
Order #154 **$6.95**

Small Business Programs **Order #156**
Complete listings for the business user. Inventory, Invoice Writing, Mailing List and much more. Introduction to Business Applications. **$14.90**

The Strongest Link

Copylink is a flexible telecommunications package with an impressive array of commands to get you on-line.

By Martin Moore

It seems that communication packages come in one of two varieties: those that are very simple, and those that are fairly complex and highly flexible. The Copylink package, from U.S. Digital, falls into the latter category.

Copylink is a communications program that takes full advantage of an autodial and autoanswer modem, such as the Hayes Smartmodem. The package is available for an entire flock of personal computers, ranging from the Kaypro and the Ferguson Big Board to the Televideo 802.

U.S. Digital promises support for the IBM PC soon, as well as a number of other third-generation machines. My review version was configured for Heath's H89 running origin 0 CP/M on soft-sectored disk drives.

A Few Nice Features

Copylink can almost be viewed as an operating system all its own. The software is capable of copying, renaming and deleting files as well as performing standard communication operations. Although the program allows all communications protocols to be set every time you use it, you can easily customize Copylink to begin execution in a state suited to your specific needs.

A couple of very nice features of the package are its ability to store telephone numbers internally and its built-in help facility.

The Copylink program can store up to ten name/number combinations. In other words, you can store up to ten names and associated telephone numbers (including area codes) within the program itself. If ten isn't enough, you can replace the names with filenames, each of which can store as many names and numbers as you like. Using the autodial and retry features of Copylink, you don't have to remember any numbers—just the name of the person you're trying to call.

Another nice feature of Copylink is its help facility. Every command, switch and flag in the program has a corresponding help message. In most cases, the help message is a single-line description of the command function. I guess what impressed me most wasn't the clarity of each description (though they are fairly clear), but the fact that they were there at all. Too many software designers have yet to realize that an on-line help facility should be an absolute requirement. The folks at U.S. Digital did a good job here.

A Commanding Set

Copylink runs two modes: command and terminal. The command mode is used to set up operating parameters prior to connection. The terminal mode is used once connection with a remote system has been established. Commands are entered in command mode as ASCII strings (ANSWER, for example) or in terminal mode as escape sequences (ESC H). The following is a list of some of the more interesting Copylink commands. Copylink also has commands that let you adjust your communications parameters and file management commands that let you rename, delete, copy and view disk files.

Active—The Active command sets the amount of time the software will wait before hanging up on an idle connection.

Add—Copylink is able to store up to ten name/number combinations as part of the program. In other words, every time you access the program, you can have ten previously defined numbers available. With the use of an autodial modem, Copylink can automatically dial a number for you if you specify the name. If a name is stored without an associated number, the software assumes that you're talking about a directory where more names and numbers can be stored.

Answer—This command configures the modem to send an answer carrier signal, which allows the computer to accept incoming calls. At default, Copylink is set to run in originate mode.

Answerback—This command is used to examine the answerback string, which is the message sent to a remote system in response to a call.

Auto—This command sets Copylink to automatically answer an incoming call.

Binary—This command allows binary transfers in terminal mode.

Blind—The Blind command turns off all error checking and allows files

Address correspondence to Martin Moore, 2735 S.W. 229, Aloha, OR 97006.

to be dumped directly from the H89 disk through the modem to the remote system. The command is used when the remote system can't perform the same kind of file transfer protocol used by Copylink.

Call—This command makes use of Copylink's ability to store names and numbers. If you have an autodial modem, you can initiate communications with a remote system by simply entering: CALL name.

Camp—If a remote system doesn't answer, you can tell Copylink to keep trying until a connection is made.

Chat—Using this command is like linking two terminals together. Whatever is entered on one terminal will show up on the other. File transfer isn't allowed when you're chatting.

Checksum—This is one of three error-checking protocols that can be used by Copylink.

CReturn—This command is used to automatically return to the Chat mode after a file has been sent or received.

Custom—The Custom command is used to store a customized version of Copylink. If, while running the program, you have it set up just the way you like it, you can store the current version in a file.

Delay—This command is used to set the amount of time Copylink will wait between sending characters to the remote system. Delay is used while the slow flag is set.

Edit—This flag allows Copylink to handle command line editing on the remote system when running in half-duplex mode.

EOL—Copylink lets you specify the end-of-line string sent to the remote system at the end of each line. The EOL command shows you the EOL string in hexadecimal format.

Fast—This sets another of the three error-checking protocols that can be used by the program. Fast is a protocol unique to Copylink and can be used for very fast communications in a local, hardwired connection.

Filter—This flag is used to limit, or filter, the number of control characters that are sent are by the remote system to return, linefeed, bell and backspace.

Macro—The Macro mode allows you to send terminal commands to the remote system from a local file.

Monitor—Enable Monitor mode. This mode allows all control characters sent by the remote system to be displayed as hexadecimal numbers.

Multiple—This flag enables or disables multiple mode file transfer (transfer of more than one file at a time).

New—This command is used to install a new disk on the system and to change the status of a disk from read only to read/write.

Normal—This is the third Copylink error checking protocol. Normal performs cyclic redundancy checking (CRC).

Printer—Enables the printer toggle. If the printer toggle is enabled, entering a ↑P (control-P) while communicating with the remote system will allow information sent from the remote system to be dumped on a printer.

Quiet—Under normal operation, file transfers to and from the remote system are echoed to the H89 screen. Quiet turns the echoing off.

Remote—The Remote command allows Copylink to accept certain commands from the remote system.

Remove—Remove names or filenames from the internal phone directory.

Save—If this mode is enabled, information received while in terminal mode will be written to the open terminal receive file.

Slow—When running in Slow mode, Copylink will insert a delay between each character sent. The delay (in milliseconds) is set by the Delay command.

TReturn—This switch will set Copylink to return to the terminal mode after a file has been transferred.

Turn—This command sets the turnaround delay used in half duplex operation.

All of these commands have their equivalents in terminal mode and are entered as escape sequences.

Floods of Documentation

Copylink comes with an almost overwhelming amount of documentation. Along with the more than 250-page user's manual comes a copy of *The Complete Handbook of Personal Computer Communications* by Alfred Glossbrenner.

The user's manual is divided into four sections and an appendix. The first section is entitled "For the Beginner." The trouble is—it isn't. I suggest that U.S. Digital either throw the section away or have it rewritten. As a user with some experience with communications packages, I found it confusing and misleading. I would guess that a complete novice would get so bogged down in the contradictions and references to unknown facts that he would give up.

The remaining sections of the manual are pretty well-done and are geared toward getting things done. For example, the manual tells you how to set up your on-line phone book, as well as how to customize Copylink to your particular needs.

Speaking of customization, Appendix B of the manual describes in detail the portion of Copylink that you can permanently change, should you so desire. Appendixes C and D contain listings of the .ASM files that you can modify.

One of the most interesting things about the manual is in Appendix A. The company placed well over 500 phone numbers of bulletin boards and electronic message systems—a truly impressive list.

Sheer Flexibility

For sheer flexibility, the Copylink program from U.S. Digital is a winner. For ease of use, it can be rated as high as any other, and higher than some. The only problem you might run into is simply remembering all the commands available. I found, however, that after I had used Copylink for a little while, I could easily remember the most often used commands.

The price of Copylink ($99.95) is right, considering the kind of capabilities you get. If you need a good, solid communications program, I recommend Copylink.■

A Capsule Look At Copylink

Manufacturer
U.S. Digital Corp.
5699-D S.E. International Way,
Milwaukie, OR 97222.
Price
$99.95
Features
Full autoanswer/autodial capabilities, any standard bit rate; built-in telephone number directory; help facility; file deletion, copying and renaming; drive selection.
Systems Supported
Kaypro, Osborne, Morrow Micro Decision, Otrona Attache, Televideo 802, Heath H89, Ferguson Big Board, IBM PC, Columbia Data, COMPAQ, Victor 9000, TRS-80 Model III, Eagle, North Star and more.

Macintosh
Apple's Saucy New Micro

The Macintosh is different, well perhaps revolutionary is a better word. So hold on to your chairs and read about how Mac will shake up the microcomputing industry.

Last month *Microcomputing* featured a preview of Apple's Macintosh, the heralded machine that was finally introduced at Apple's annual meeting on January 24, 1984. Since the Macintosh preview I've had the opportunity to test the machine in *Microcomputing*'s Peterborough office. I must note that while the Macintosh I used was a production machine, the software provided had not yet been finalized.

Not Your Ordinary Micro

If you've been using micros for some time, you'll find that Macintosh challenges the usual concepts of operating a microcomputer. In fact, neophytes might even have an easier time using the machine since they have never had to deal with cumbersome operating systems and their related peculiarities. You'll marvel at how transparent an operating environment can actually be.

Nowhere in the Macintosh operating manual is there a single reference to operating systems, DOS, PIP or any other of the jargon that has left many potential computer users baffled.

Instead, Macintosh is described in an easily understandable vocabulary of metaphors. To fully understand the extended metaphors used for Macintosh, you should keep a finger in the manual's glossary at the book's rear for frequent reference.

Most of the concepts used to describe Macintosh were introduced when Apple released Lisa in January 1983.

Since then, the Lisa's windowing technology has been the object of intense competition among most of the largest software publishers. In recent months, software imitating Lisa technology has been introduced or announced by Microsoft (Windows) and Visicorp (VisiOn).

To clarify the concept of windows, let's agree that they display information on a screen that is likened to a desktop. You can view documents, edit or change them, scroll through them and move them around on the "desktop."

Rather than read and select numerous menu items, you need only to indicate a document or application represented pictorially (by an icon) and the computer will open a document and the accompanying application program. The basic concept of windows will provide us with a departure point for a further understanding of Macintosh.

The Size Surprise

One of the surprising things about

By Keith Thompson

Macintosh is its size. Within a single case are the nine-inch diagonal monochrome screen, 3½-inch disk drive and two circuit boards (the analog board with power supply, speaker, and so on, and the digital board containing the actual computer). The keyboard and mouse are the other two standard appendages, although optional disk drives, numeric keypads, modems and printers may be added.

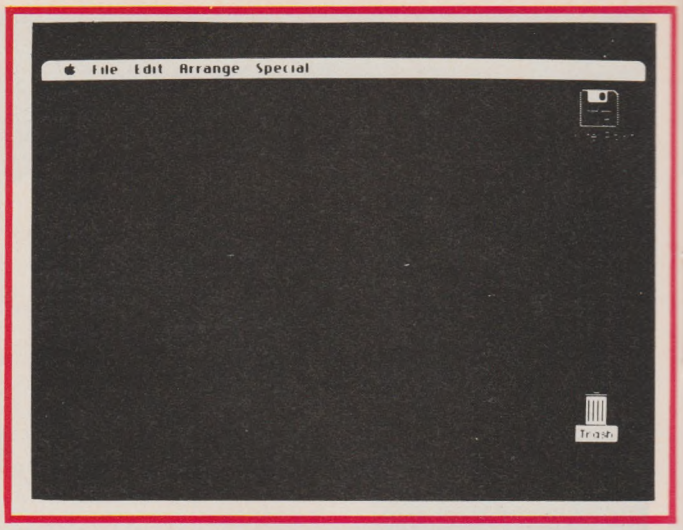
The basic Macintosh occupies no more table space than a piece of paper. It resides unobtrusively on a desktop. Its detachable keyboard snugs up under the machine when not in use. The main unit is made of high-impact plastic in Apple beige. Housed within is a machine that may revolutionize the microcomputer market.

Within the desktop window there are a few concepts that you must first familiarize yourself with. The mouse is the machine's pointing device. While some actions of the mouse can be implemented using the keyboard, the mouse is an integral piece of this machine. A one-button control makes the mouse easy to operate. The command structure for the mouse is simply point, click, click-click or (in rare cases) shift-click. Learning these commands is a short process. The mouse track requires at least five inches across and 3½ inches down to move the cursor around the screen from edge to edge. Realistically, though, a square foot or more is necessary to provide enough room to move the mouse without your having to keep picking it up and putting it down.

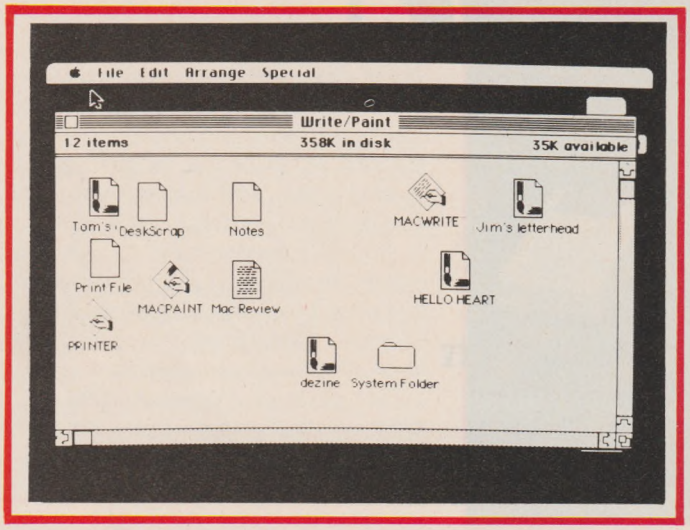
The keyboard is connected to the front of the main unit (why do so many manufacturers insist on plugging in their keyboards from the rear?). The cord is coiled with telephone-type modular plugs on each end. It extends to a little more than three feet. However, having it so far from the mouse would preclude a laptop keyboard use for many applications. The keys have a solid feel to them—more solid than most micros have been offering. There is no key-click option, but the distinct motion of each key pressed diminishes the need for an audible signal.

The basic keyboard layout is standard. There are, however, some strange conventions that may take some getting used to. First, there is no control or escape key. Interfacing the Macintosh with another computer may be unwieldy, even assuming that the codes can be emulated otherwise.

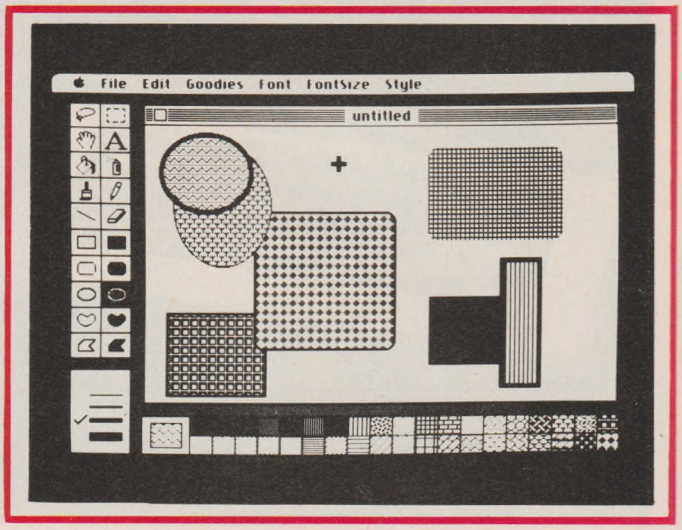
Address correspondence to Keith Thompson, c/o Microcomputing, 80 Pine St., Peterborough, NH 03458.



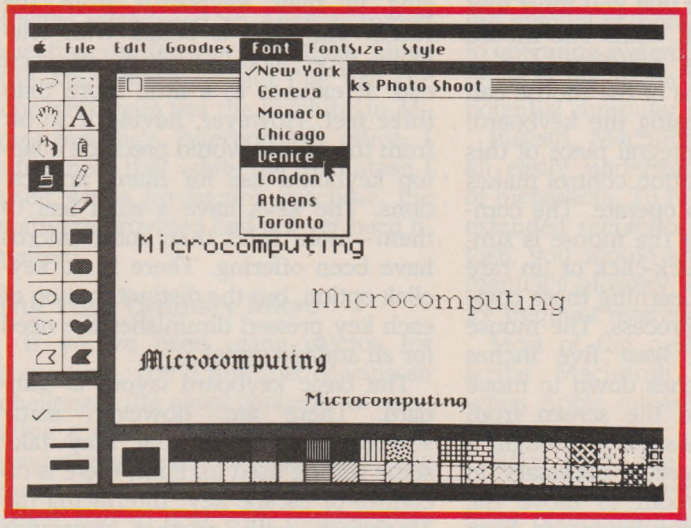
Disk icon showing what disk was read by Macintosh. There is a trash can to dispose of unwanted documents. Note the menu bar at the top of the screen.



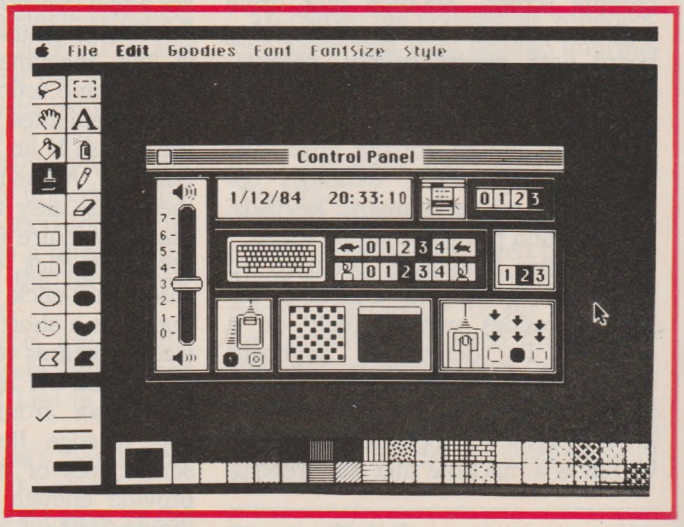
Pointing the cursor to the disk icon and clicking with the mouse activates disk window.



Mac/Paint screen showing paint options on the left side and pattern choices along bottom of desktop.



A photo of a sample MacPaint screen. The font menu has been pulled down, which allows you to change fonts within the text.



MacPaint's control panel lets you change system defaults, such as speaker volume, repeat key rate, cursor blinking light and so on.

There are two keys on each side of the space bar—an option key on each side, a command key (labeled with a strange little design) and an enter key that is slightly different in function than the larger return key that resides in its normal place above the left shift key.

As mentioned above, the main unit contains a single 3½-inch micro disk drive. The media used is the Sony microfloppy, the same media that Hewlett-Packard committed itself to last year with its HP-150 Touch Screen computer. For those who may not be familiar with the 3½-inch microfloppy, a hard plastic case (⅛-inch thick) protects the magnetic media contained within.

The front side of the disk contains a metal shutter that remains closed when not in use. Inserting the disk into the computer opens the shutter, making the disk accessible to the disk drive head. One of the nice features of this media is a sliding tab that functions as a write-protect tab. Should you decide to change the protect status of a disk, you need only slide the tab. A hole in the disk envelope gives quick visual indication of whether or not the disk is write-protected. The choice of this particular media by two major manufacturers may quicken the fate of the other micro disk standard contenders, such as Dysan who has been trying to gain acceptance for its 3¼-inch soft-envelope disk.

One fact about the Macintosh that is sure to evoke criticism is its lack of internal expansion. In fact, contrary to Apple II philosophy, the machine is sealed with a warning not to remove the cover. Unlike the Apple II, though, the Macintosh comes equipped with a built-in clock, two serial ports and an external mass-storage connector. What's in it now is what users and programmers will have to work with. The machine has the Motorola 68000 32-bit chip at its heart. The chip runs at 8 MHz—allowing it to handle its complex windowing tasks in a relatively short period of time. The 64Kb of ROM contains the framework for the machine's operation. There are more than 500 calls that can be made into the ROM to manage the details of the window displays and pull-down menus and to organize disks and documents.

Contained within this ROM is what Apple calls the "Finder." This Finder is, for most intents and purposes, the built-in operating system. It takes care of opening, closing, copying, discarding, moving and renaming documents. It is also responsible for organizing documents on the desktop, in folders and on disks as well as initializing and ejecting disks.

Stepping Through Mac

Perhaps the best way to understand the workings of this unorthodox machine is to step through the process of turning on this machine and running a typical application program.

The ac switch is located in the right upper rear. Rocking the switch initializes the machine and performs a quick self-diagnostic. A beep indicates that the machine is working properly. Shortly, a disk icon appears in the middle of the screen. It has a blinking question mark in it as it awaits your choice of disk. Once a disk is inserted, the Macintosh checks it for readability. If it is readable, a "smiling Macintosh" icon appears while the disk read is being completed. As it is loading—in fact when there is any waiting—the mouse cursor transforms itself into a small wristwatch.

When the disk has been loaded into Macintosh, an icon appears on the right side of the desktop. A trash can appears on the bottom right of screen. Along the top of the screen is the menu bar.

The mouse pointer can now be positioned atop the disk icon and clicked once. The drive activates and within a few seconds the contents of the disk are displayed as icons in the window. It is easy to identify which files belong together—each is graphically labeled to identify it with its respective application program. For instance, designs saved by MacPaint are depicted by title below a small pallette; spreadsheet documents are seen as miniature spreadsheets and so forth. If you prefer some traditional methods of viewing, moving the mouse cursor and selecting the View option from the menu bar will allow you to arrange the disk contents by name, kind, date modified or size. One of the intriguing features of Macintosh's windows is the method of moving documents around the window, desktop or from disk to disk. The term used is "dragging."

To rearrange the disk directory screen, the cursor is placed over the file to be moved. The mouse is clicked and held, allowing the icon to be moved to anywhere on the screen. Files that are related can be dragged to a folder icon. It is here that they are stored together. The analogy is to a manila folder where you might keep a set of budgets created on a spreadsheet program. When you want to see the folder, you point to it and click. The folder immediately opens, showing the contents. This keeps the desktop clean and allows you to keep your Macintosh files organized by subject matter—much like we attempt to do on our real desks!

Take a Letter

If you wanted to type a new letter, you would position the mouse cursor over the word processor application icon and click once. The disk drive would activate and, in a few moments, the word processor would be fully functional on the screen before you. Across the menu bar at the top of the screen are seven menu selections.

One of the strongest points of the Macintosh is its horizontal menu structure; from application to application the process of loading, saving and otherwise manipulating data within different programs remains essentially the same. This is made possible by the extensive ROM architecture that allows programmers to exploit the Lisa technology with a minimum of adaption from traditional programming design techniques (see sidebar). For the programmer, this saves time and encourages program adaption to Macintosh. Once you master the basic technique of one program, other programs will be structured in a similar manner.

While you are in any application, you can choose the Apple menu (it's the left hand Apple logo on every menu bar). This is referred to as the Desk Accessories menu and, quite frankly, this item, besides being practical, is a lot of fun. On-line at all times is a calculator, clock, key caps (described later), note pad, scrapbook, control panel and puzzle.

Selecting the calculator brings a calculator to the desktop. This calculator will function using the mouse pointer to depress keys or it will work by typing the numbers directly from the keyboard. It is handy when, in the middle of a letter, you need to add up a column of numbers. The calculated answer can then be directly input into an application document.

Another selection is the clock. The Macintosh has a real-time clock as standard equipment. The clock can be evoked and used to date a memo, left on the screen to keep a watch on the time or, as Macintosh itself uses it,

date all file activity.

The preliminary documentation mentioned being able to set an alarm, although I couldn't find the way to do it. Key Caps answers the problem of which keys to what when the option or command key is pressed. With this option, a keyboard is displayed on the screen. Each key is labeled as to what character it generates. When a key, such as the option key, is pressed, the alphanumeric illustration on the screen changes to show what a particular key displays.

Incidentally, the number of characters displayed is impressive. Aside from a number of extras (the "tm" and "reg" and copyright symbols), you will find scientific, mathematical, Greek, foreign and diphthong representations.

If the proper tables were constructed, the Macintosh could easily input directly into phototypesetting equipment using the special typographical symbols available. The Notepad selection displays a pad of paper. If, in the middle of a program, you wished to jot down a message, you only need pull down the note pad, write your note, put it away and continue with your work. The notepad has eight sheets that can be turned by clicking the mouse on the corner of the pad. The Scrapbook allows you to keep pictures and text that you use frequently. By "pasting" them in the scrapbook, you can select and use them in many of the other application programs. For example, if you were to design a letterhead using MacPaint, you could paste the letterhead in the scrapbook. Then, when preparing to print out a spreadsheet, you could paste the letterhead at the top of the printed page. A little imagination will yield many uses for this one.

The Control Panel fills most of the screen with graphic representations that, with the use of the mouse, will let you set your speaker volume, repeating key rate and even background pattern of your desktop. Most of the settings will be remembered when the system is powered down.

Perhaps the most fun item on the Apple menu is the Puzzle. This is the kind with 15 numbers in a 4×4 grid numbered from 1 to 15. With the mouse, you rearrange the pieces in numerical order. The puzzle rescrambles whenever you close it.

A Window on New Technologies

It is intriguing to exploit the window technology. Along each window are two arrows—one at the top right pointing upward and one at the bottom right pointing downward.

By placing the cursor over one of these, you can click the mouse and scroll the document up or down. In the extreme lower right corner of the window there is the size box, which changes the amount of space the window occupies on the screen. By pressing the mouse button on the size box and moving the mouse around the screen, the window can be sized in either direction. Whatever size the window is on the screen, the contents of the window don't change. The only change is in what you see. Entire windows can be easily moved around the screen by pressing at any position in the window's title bar. The window can then be moved aside on the desktop until needed.

With variable size windows, you can open more than one file, view it, and cut and paste information from one document into another. Only one window may be active at any one time, but any inactive window can immediately become active by merely clicking the cursor at any place in it.

When you have completed work on your document and are ready to save it, position the cursor over the File menu, pull down the menu and select "save as. . .".

The Finder asks you for a filename, then the document is saved. If you wish to make frequent saves while working on a document, just position the cursor on another option, Save, and Macintosh will store your revised document on disk as you continue to work on it.

Choosing quit from the File menu will remind you to save the final version of your document to disk and return you to the desktop, which shows the entire contents of the disk. Opening any document you wish to work with also activates the application you need to do the work.

Any Comments?

It might be wise to make some comment about each document you create. To leave a comment about a particular file, select the icon that represents your document. Then choose Get Info from the File menu. An information box appears showing which disk the document resides on, its size and, at your option, whether the document should be locked to prevent accidental deletion.

Within this window is space for you to type two lines of comments about the document. This information box is quickly available whenever you are in the application disk window, which saves you the time of opening many documents to find the one you need.

Another function that is available in every application is cut and paste. Macintosh contains a clipboard that stores information that is cut or copied from a document. So something cut from one document can be easily moved to another by opening the other document, positioning the cursor to the proper location and pasting. This works with graphics as well as with passages of text. A similar, but separate, function is available with the "scrapbook." This desk accessory has a larger capacity than the clipboard and allows more storage for graphics, boilerplate paragraphs or letterheads.

Showing Off Its Windows

In January, Apple introduced MacPaint and MacWrite, a graphics and word processing package that shows off the window technology of Macintosh.

Microsoft also introduced Multiplan—this application alone could do for Macintosh what Visicalc did for the Apple II several years ago.

Several software publishers are developing or adapting application packages for the Macintosh. Microsoft is planning to introduce versions of Chart, Word and File during 1984. Barney Stone of Stoneware thinks that Advanced DB Master may be available in the second or third quarter of this year. Lotus has also confirmed that they are adapting 1-2-3 to the Macintosh environment. Software Arts is working on a release of its pfs series and Think Tech is developing Pascal for the machine. Living Videotext is adapting its ThinkTank for introduction in the near future.

Apple had selected about 75 software developers to support in their efforts to program for Macintosh. Most of them didn't have Macintoshes. In fact, they developed their software in the Lisa environment, then adapted it to Macintosh. Rapid acceptance of Macintosh by software publishers is vital to the longevity and success of the machine. The introduction of significant application packages will give the Macintosh an impetus never realized by the Lisa.

Apple Steps Aside

Apple has made it clear that after it introduces nine fundamental programs

DOUBLEDOS

Makes your IBM PC twice as good for only $299!

Your IBM PC can perform even better than IBM intended. Introducing DoubleDOS—the SoftLogic Solution that lets you run two programs at once—with no down time. Who said you can't do two things at once?

This magical solution to better productivity is introduced at only $299. ***Plus—there's no need for expensive additional hardware or software changes in your PC.*** DoubleDOS works with DOS 1.1 or 2.0, and your existing DOS software.

To order DoubleDOS or for more information, call SoftLogic Solutions toll free at 1-800-272-9900, or mail in the coupon below immediately. Ask for DoubleDOS at your local personal computer dealer.

DoubleDOS—and the IBM PC—a solution that's twice as sweet as a rose!

SOFTLOGIC SOLUTIONS
530 CHESTNUT STREET
MANCHESTER, NH 03101
1-800-272-9900
In NH call 627-9900

DoubleDOS works with DOS 1.1 or 2.0, and your existing DOS software.

Circle 271 on Reader Service card.

ORDER NOW!
Rush me DoubleDOS.

☐ I'VE ENCLOSED $299 (CHECK OR MONEY ORDER)
☐ CHARGE TO MY VISA/MASTERCARD (CIRCLE ONE)
ACCOUNT NUMBER: _____
EXPIRATION DATE: _____ M
SIGNATURE: _____
NAME _____
STREET _____
CITY _____ STATE _____ ZIP _____
SEND TO: SOFTLOGIC SOLUTIONS, INC.
530 CHESTNUT STREET • MANCHESTER, NH 03101

MasterCard VISA

for Macintosh it wishes to stand clear for third-party software developers to really support the machine. The bulk of programs from Apple will cater to the programmer—MacPascal, MacBasic, MacAssembler/DeBugger and MacLogo. In addition to the previously introduced MacPaint/MacWrite, MacTerminal (for use with the Apple modems), MacProject and MacDraw will be released. Most of the progrms will cost less than $100.

Apple's Peripheral Vision

The rear of Macintosh contains three I/O ports that support external interfacing activity. There are two similar RS-232C/RS-422 serial connectors. One of these is identified by a printer icon. As of this writing, the only printer supported by the Macintosh is the recently introduced Apple Imagewriter dot matrix printer.

Imagewriter is capable of an exact reproduction of Macintosh graphics and text. Hopefully the printer bus will be adaptable to other serial printers. While connection to other printers will undoubtably sacrifice the graphics option, a letter quality printer will be required for business correspondence. Apple is developing a laser printer that could give excellent quality for correspondence while allowing the Macintosh graphics capability to be exported to paper.

Another RS-232C/RS-422 connector is labeled for modem use. Apple now offers two modems that, with MacTerminal software, will give the machine full telecommunications capabilities. The terminal software will, with an additional hardware accessory from Apple, emulate IBM 3270 terminals. Tecmar has also announced an intelligent modem for the machine.

The third I/O port is reserved for external mass storage capability. Apple is offering a second external disk that is identical to the 400Kb internal drive. Tecmar has announced two products to expand Macintosh's storage capacity.

Apple will be offering a five-megabyte hard disk with a removable cartridge. In addition, to compensate for delays while printing, it will be marketing a RAM spooler for the printer. The spooler will be combined with a 512Kb disk emulator. The convenience of the spooler and speed of the emulator make this add-on worth serious consideration.

Another significant option is being offered by Tecmar; an IEEE-488 interface. With this interface, you'll have the ability to access the large number of scientific instruments that are compatible with this bus. You will be able to interface the catalog of Hewlett-Packard products that use this bus as their standard.

Wait and See

There is one port that is not mentioned in the documentation I received. I wasn't able to locate anyone at Apple who could explain this plug, labeled "4 channel sound port." The connection is common enough—it accepts a regular mini-audio jack. We will have to wait to see about this one.

Tecmar, as of this writing, has not released prices on its peripherals. In a future issue, *Microcomputing* will be reviewing the hardware products mentioned along with price information.

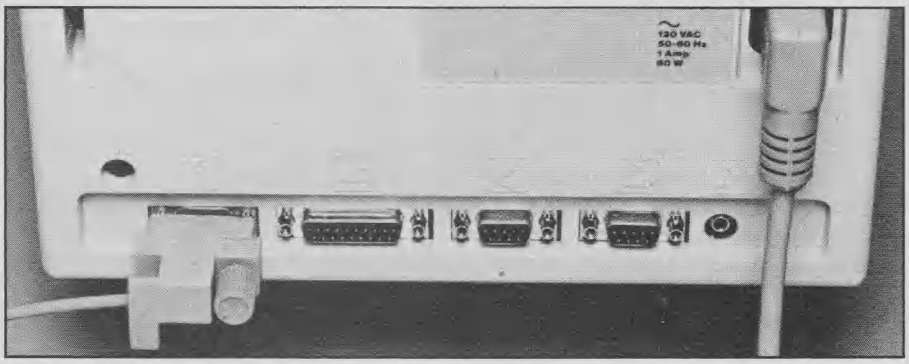
A New Breed of Micro

Macintosh is a machine that is fun. As an Apple II owner, I felt uncomfortable and even suspicious about a computer that is so easy to operate. Missing was the familiar Applesoft prompt—a PR#6 before reading a disk.

What I found was a machine that was so easy to operate that my eight-year-old would constantly pester me to use the Macintosh while I was writing this review. Within an hour she was well-versed in the terminology and in how to use the machine. This type of ease of use may create some resentment in the hacker in you—the machine consciously places distance between you and the operating system. In the Macintosh case, the term user transparent would be more accurate that user friendly.

Macintosh represents state-of-the-art windowing technology at a price you might pay for just window software for the IBM PC. Its speed and sophistication place it above any window products I've seen for the PC. Its size marks it a welcome appliance on a desk.

Plans for integrating Macintosh into a networked office system containing a Lisa at its heart and Macintoshes on most desks will appeal to almost anyone who, in Apple terminology, is a "knowledge worker." Whether or not you're in the market for a computer, experience a Macintosh. I bet that you will want one!■



The I/O ports of the Macintosh showing (from left to right) the connectors for the mouse, external storage, modem, printer and four-channel sound.

A Capsule Look At the Macintosh

Manufacturer: Apple Computer, Inc., 20525 Mariani Ave., Cupertino, CA 95014.
Dimensions: Measures 13.5 × 9.7 × 10.9; weighs 16 pounds, eight ounces.
Processor: MC68000, 32-bit architecture, 7.8336 MHz clock frequency.
Memory: 128Kb RAM, 64Kb ROM
Disk Capacity: 400Kb preformatted disk; 3½-inch diameter hard-shell media.
Screen: Nine-inch diagonal, high-resolution 512-pixel by 342-pixel bit-mapped display.
Interfaces: Synchronous serial keyboard bus; two RS-232C/RS-422 serial ports, 230 4K bits per second maximum (up to 0.920 megabits per second if clocked externally); mouse interface; external disk interface.
Sound Generator: Four-voice sound with 8-bit digital-analog conversion using 22 KHz sample rate.
Input: Line voltage: 105 to 125 volts ac.RMS
Frequency: 50 or 60 Hz
Power: 60 watts
Keyboard: 58-key two-key rollover, software mapped; measures 2.6 × 13.2 × 5.8; weighs two pounds 8.5 ounces.
Mouse: Mechanical tracking, optical shaft encoding 3.54 pulse per mm (90 pulse per inch) of travel; measures 1.5 × 2.4 × 4.3, weight seven ounces.
Clock/calendar: CMOS custom chip with 4.5V (Eveready No. 523 or equivalent) user-replaceable battery backup.
Price: $2495; Imagewriter costs $495 (if purchased with Macintosh).

Quickport and Toolkit/32: Two Keys to More 32-bit Software

One of the biggest criticisms of the original Lisa was that there was very little software available for it. You either bought the Lisa application series from Apple or you didn't have software.

Apple heard the criticism—through its dealers, through Lisa owners and through sales figures. The company's answer? Quickport and Toolkit/32. Both are designed to get more programs running on electronic desktops.

Toolkit/32 lets software developers create integrated software that takes full advantage of the desktop and windowing environment. Apple calls it a "generic application"; that is, all the program code required to support the mouse, create windows, display pull-down menus, use cut-and-paste features and so on is already provided.

The developer's job is simply to add the features that make the application unique. For a developer writing a database manager, that means adding sorting and report-generating routines (to name only two); for someone developing an accounting system, it means adding the number-crunching routines and some type of report generator.

This approach makes program development for the Lisa/Macintosh environment as easy as, if not easier than, program development for any other computer. Programmers don't need to know how to write window or cut-and-paste routines—all that has been figured out for them. All they have to do is fill in the blanks to end up with an application that looks as sexy as any Lisa/Macintosh application.

Semicustom Generic

Bruce Blumberg, manager of third-party products and development tools for Apple, compares Toolkit/32 to a semicustom-built house, where the builder supplies the foundation and framework and the designer determines the rest—color, decor and special features.

The generic application is based on Clascal, an Apple-developed superset of Pascal. Clascal is an object-oriented language. This means that aspects of a program—the menus, windows and so on—are described as objects. When writing a program in Clascal, you first define the object and then specify how the objects interact with each other.

The real power of Clascal lies in the fact that objects can automatically inherit the general characteristics of previously defined objects.

Clascal descriptions of all the objects used in the Lisa/Macintosh environment are inincluded in generic applications. These descriptions are inherited by applications developed under Toolkit/32.

Toolkit/32 also contains what Apple calls "building blocks", which are Clascal descriptions that contain the Lisa/Macintosh basic graphics, text editing and dialog functions. Building blocks can be used by developers to create graphics, word processing capabilities and messages for their programs.

According to Blumberg, Toolkit/32 has been praised by software developers anxious to climb aboard the electronic desktop. To give them the final boost, Apple is offering one-week training classes to qualified developers. Graduates of the $1200 course receive Toolkit/32 software and documentation and periodic updates plus personal support from a qualified engineer for one year.

Quickport

Quickport, soon to be available for Lisa, provides an even easier way to get programs running under the Lisa environment. Quickport lets you adapt programs written in Pascal, Cobol, Basic Plus and other languages to the Lisa desktop. Once adapted, a program has its own window, icon and stationery pad (used for creating documents).

You can then scroll, resize, open and close the window, use the same print functions as the Lisa applications, cut and paste between applications and have any number of windows open on the desktop. When running, the programs themselves will look much like they did before being adapted.

Quickport is most useful for programs that don't need the pull-down menus and graphics provided by the Lisa environment. It'll be especially valuable to companies that are running (and relying on) existing programs but want to be able to use them on their new Lisas without having to do any rewriting.

Take a Step

Quickport and Toolkit/32 show that Apple is taking steps to remedy one of the biggest problems the Lisa faced—lack of software. With the addition of Lisa 2 and Macintosh to their 32-bit family, and with the interest shown by the 75 or so developers now working with the machines, it's safe to say that Apple's 32-bit software blues will soon be a thing of the past.

J.H.

A Glossary Of Terms For Macintosh:

Active Window: The foremost window on the desktop, the window where the next action will take place.

Click: Position the pointer with the mouse and press and quickly release the button.

Clipboard: The temporary holding place for what was last cut or copied.

Control Panel: A desk accessory that lets you change speaker volume, the keyboard delay and repeat speed, and other preferences.

Cut: To remove something by selecting it and choosing Cut from the menu.

Desk Accessories: "Mini-Applications" that are available from the Apple menu regardless of what application you are using. Examples are the Calculator, Note Pad, Clock and Puzzle.

Desktop: Macintosh's working environment—the screen display.

Dialog Box: A box containing a message requesting more information from you.

Drag: Position the pointer on something, press and hold the mouse button, move the mouse and release the button. When you release the mouse button, you either confirm a selection or move an object to a new location.

Finder: An application that's always available from the desktop. You use it to manage documents and applications and to access disks.

Folder: A holder of documents and applications on the desktop. Folders allow you to organize information in any way you want.

Icon: A graphic representation of an object, a concept or a message. Icons are often messages associated with the finder.

Menu: A list of commands that appears when you point to and press the menu title in the menu bar. Dragging through the menu and releasing the mouse button while a menu item is highlighted chooses that item.

Paste: Used to reposition the contents of the Clipboard—whatever was last cut or copied.

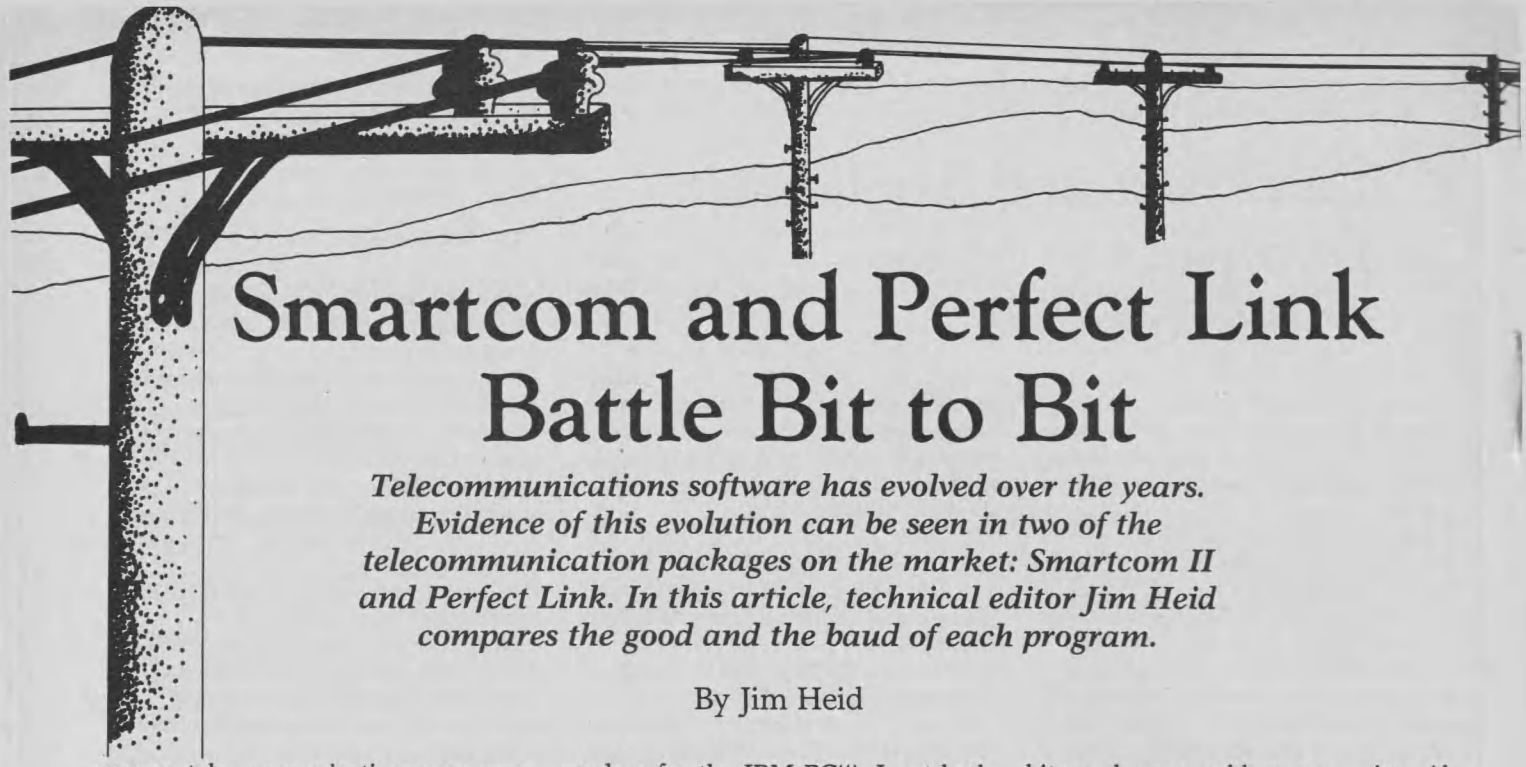
Scrapbook: A desk accessory where you save frequently used pictures or passages of text.

Scroll: To move a document or directory in its window so that different part of it is visible.

Scroll Arrow: An arrow on either end of the scroll bar. Clicking a scroll arrow moves the document or directory one line. Pressing a scroll arrow scrolls the document continuously.

Scroll Bar: A rectangular bar that may be along the right or bottom of a window. Clicking or dragging in the scroll bar causes the view of the document to change.

Size Box: A box on the bottom right of some active windows that lets you control the size of the window.



Smartcom and Perfect Link Battle Bit to Bit

Telecommunications software has evolved over the years. Evidence of this evolution can be seen in two of the telecommunication packages on the market: Smartcom II and Perfect Link. In this article, technical editor Jim Heid compares the good and the baud of each program.

By Jim Heid

Boy, telecommunications programs sure have come a long way since I was a kid. It used to be that your terminal program was smart if it could merely send data to your printer as it was received. If you didn't print a hard copy of your communication, whatever information you exchanged vanished as the screen scrolled. If your printer was slow, you lost the first word or two of each line. Add to this the burden of user-hostility—no menus and cryptic control-key sequences—and telecommunicating became a chore, not a pleasure.

All that has changed. A new class of really sophisticated telecommunications packages is on the market, thanks to a few new developments. Computer memories are growing because of lower chip prices and 16-bit CPUs. Information services and bulletin boards proliferate. Modem prices are falling, putting modems in more hands and making telecommunications programs lucrative pursuits for software developers.

You're Not a Kid Anymore . . .

Perfect Software's Perfect Link and Hayes Microcomputer Products' Smartcom II are two recent entries to the new class. Both are powerful packages that make me glad I'm not a kid anymore.

I took a long look at the IBM PC versions of Perfect Link and Smartcom II. I tested both programs on an IBM PC and on a Compaq Plus, under both DOS 1.1 and 2.0. The modem used is the new Hayes Smartmodem 1200B (see the sidebar "A Plug-In Smartmodem for the IBM PC"). I put both programs through their paces for about two months, using them to access The Source, CompuServe and numerous bulletin board systems. Both performed well, but Smartcom II came out on top. It has an elegance that Perfect Link lacks; it seems better thought-out, has far more on-line help and is easier to use.

Features on Parade

The Capsule Look lists what each system needs to run and what each costs. Perfect Link will be available soon for CP/M and other MS DOS machines; Smartcom II is also available for the Wang Professional, the Texas Instruments Professional and the DEC Rainbow.

Like just about every telecommunications package, both programs let you adjust the communications parameters. Perfect Link has Smartcom II beat here. Smartcom II supports bit rates of 110, 300 and 1200 bits per second (bps); Perfect Link supports 110, 150, 300, 600, 1200, 2400, 4800 and 9600 bps operation. Because Perfect Link supports bit rates up to 9600 bps, you'll probably find it more useful than Smartcom II for file transfers between hard-wired computers.

Besides bit-rate adjustments, both programs let you define word length and data format. Perfect Link is again more versatile. It supports five- through eight-bit words, with one or two stop bits and even, odd or no parity.

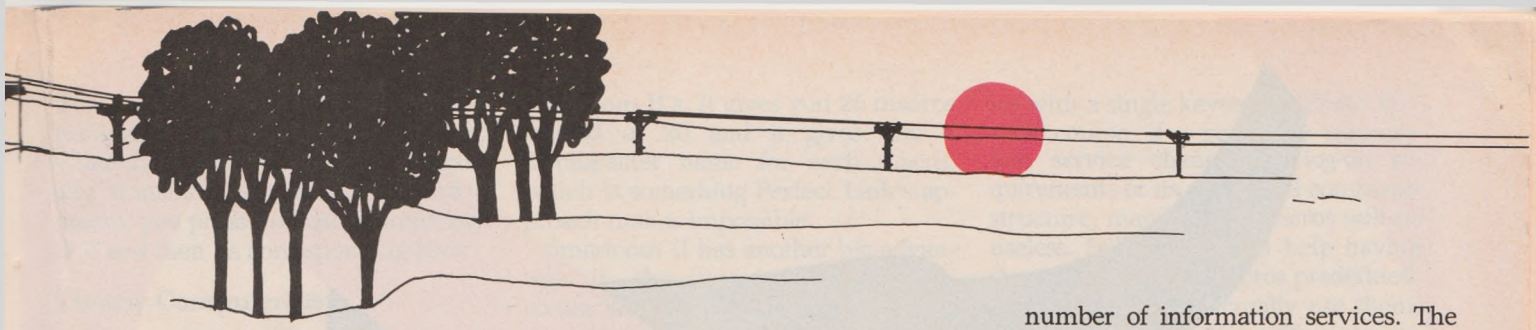
Smartcom II supports seven- or eight-bit words, with one or two stop bits and even, odd or no parity. Although Perfect Link is more versatile in this department, most telecommunication takes place using seven- or eight-bit words, so the extra versatility probably won't mean much in your day-to-day operation.

Both programs support intelligent modems like the Hayes Smartmodem, but Smartcom II supports them a little better. While both can automatically dial a number using pulse dialing, Smartcom II also supports touch-tone dialing. You can also tell Smartcom II to redial a number until it hears a carrier tone; Perfect Link has no such feature. Smartcom II can also automatically answer the phone, and you can specify that it answer after a certain number of rings (up to 255).

Unfortunately, neither program uses the system clock to support unattended dialing. This means that you can't tell either program to dial and access an information service during the middle of the night, for example, when rates are low. A lot of less sophisticated programs support unattended dialing; it's disappointing that these don't.

Each program does provide a clock display. Smartcom II displays the system time and date at the bottom of the screen, while Perfect Link's clock resets to 00:00:00 and starts counting when you go on-line (Photos 3 and 4). I

Address correspondence to Jim Heid, Microcomputing, *80 Pine St., Peterborough, NH 03458.*



like Perfect Link's approach here. It's convenient to be able to glance at the screen and see how long you've been signed-on; it helps you to judge how much a telecommunications session is going to cost.

Both programs will support your printer. If you see something you want to print, one keystroke routes it to your printer.

The programs have print buffers that prevent data from being lost if your printer is slow. Besides printing data as it comes in, Smartcom II can also print reports describing each communications and macro set (described later), and it can print a disk file.

File Transfer Features

Both programs let you transmit and receive files using a number of protocols. Smartcom II calls its protocols stop/start, send lines and verification. Stop/start is the protocol you'll probably use most. It lets the system receiving the file stop and start the transfer by transmitting a stop character and a start character to your system.

You use the send lines protocol when sending a file to a system that processes data on a line-by-line basis, as some bulletin boards do. You can use the verification protocol to transfer text and program files between two microcomputers, as long as the other computer's telecommunications program supports the Hayes verification protocol (Hayes' software does, as does M.I.T.E. from Mycroft Labs). The verification protocol is slower than stop/start and send lines, but ensures error-free transmission.

Perfect Link uses the popular Xmodem error-checking method and lets you send and receive files with or without a filename. Transferring a file "with name" means that the file's name is transferred along with its contents (assuming that the other computer system is running Perfect Link or a program that supports this kind of transfer). You can also send more than one file by using wildcard characters in the filename. When you send a file with error checking, Perfect Link waits until the other computer signals that it's ready to receive the file.

Perfect Link also lets you send a file without error checking. You use this transfer method to send a letter on The Source or CompuServe. When you send a file without error checking, the receiving computer treats the data as if it were coming from your keyboard.

Perfect Link has some additional file-transfer features. It tells you approximately how long a file will take to transfer and it shows you how much time has elapsed since you started the transfer.

Both programs also let you capture information on disk as you're receiving it, but Smartcom II's method is easier. Assume you see something on the screen that you want to save. With Perfect Link, you must first press the F1 key, which selects an option called begin/stop saving to file. The program asks you for a filename, then starts storing the data. To stop the saving process, you hit F1 again and the program asks you to confirm your choice.

To capture something with Smartcom II, you simply press the F4 key. The program immediately opens a file called Temp and begins saving to it. To temporarily suspend the capture, hit F4 again. You can restart the capture by pressing F4 once more or you can end the capture process by pressing F1.

If you press F1, Smartcom II closes the file Temp and makes you rename it. It's easier to provide a filename after you're done capturing rather than before. The ability to suspend and restart the capture process is a nice feature, too.

Both programs have file-management options that let you view directories and delete, rename and list files. Smartcom II has an additional feature that lets you create a file. The program has its own simple line-oriented text editor that you can use to create letters or messages for later transmission.

Stored Parameter Features

Perfect Link and Smartcom II let you store certain parameters for a number of information services. The programs can store two kinds of information: communications parameters and macro key assignments. The communications parameters specify how communication will take place (bit rate, data format, protocols and so on). Macros are character strings assigned to a particular key that you can recall simply by pressing the key.

These convenience features eliminate the need to reconfigure the program every time you want to communicate with a system that uses different parameters. They also let you store commonly used commands or phrases and recall them with one or two keystrokes.

Perfect Link can store up to ten parameter sets, each having its own name up to eight characters long. Each set stores information about the bit rate, data format and protocol of a given system as well as the macro key assignments. Each set can store 20 macros—each function key pressed along with a shift key, and each function key pressed along with the control key.

A macro holds up to 80 characters and can contain special characters that cause a pause, issue a break sequence, wait for particular character before continuing or denote a remark that appears on the screen but isn't transmitted. You can also specify one key as an "auto exec" key that automatically transmits its contents whenever you invoke that particular parameter set.

After you've defined a set of macro keys, Perfect Link can store them on disk for later use and revision. The program also has options that let you display the current macro key assignments.

Smartcom II can store up to 26 parameter sets—you can customize all but one. Each set has its own name, which can be up to 20 characters long. Each set can store 25 macros, and each macro can contain 768 characters. As with Perfect Link, macros can contain special information that tells the program to pause for a certain time or character. Also like Perfect Link, one macro is used for automatic log-on.

Each macro has its own name, up to 20 characters long, and a label, a letter

PALANTIR™ WORD PROCESSING

WE DON'T HAVE TO BEEF UP OUR GUARANTEE WITH A LOT OF BULL

Other software companies give you disclaimers. Palantir gives you a real guarantee: Palantir backs its software with 90 days of free phone support via a toll-free number. When you call, we won't tell you to ask your dealer or read your manual; we'll answer your question, free of charge. If we can't solve your problem, we'll replace your Palantir Software with any competitive software of comparable value.

We couldn't make an offer like this if we weren't confident about Palantir word processing. It's easy to learn, easy to use, easy to live with.

Palantir word-processing software is designed for microcomputers. Yet it gives you all the features of a dedicated word processor. You won't find a better system on the market today.

To find out more about Palantir software, call toll-free: 1-800-368-3797. In Texas, call 713-520-8221.

We'll respond with detailed information on Palantir software and a free "No Bull" button. A closer look will convince you that we're not just one of the herd. And that's no *bull*.

Circle 263 on Reader Service card.

Palantir Software 3400 Montrose Blvd. Suite 718 Houston, Texas 77006

™ Palantir is a trademark of Palantir, Inc.

from A through Z. The macro name reminds you of the macro's contents, while the label is what you use to actually transmit the macro. To send a macro, you press the macro prefix key (F5) and then its corresponding letter.

Choosy Computerists . . .

As you can see, the two programs take a very different approach toward macros. Perfect Link assigns them to function keys, which lets you send a macro by pressing control or shift and the appropriate function key. Smartcom II assigns its macros to alphabetic labels—you send a macro by pressing F5 followed by the macro's label.

Which approach is better? I like Smartcom II's. It gives you 26 macros instead of 20 and it gives you a 20-character name for each macro, which is something Perfect Link's approach makes impossible.

Smartcom II has another big advantage in the macro department—it comes with 14 of its 26 communications sets preprogrammed. There's two sets for The Source (Telenet and Tymnet), two for CompuServe, two for Dow Jones, one for Knowledge Index and several for the larger bulletin boards (ABBS in New York and Chicago, Forum-80 in Kansas City and others). The sets for the big information services even have preprogrammed macros that let you access specific topics with a single keystroke (Photo 1).

Of course, if a particular information service changes its log-on requirements or its menu and command structure, many of the macros will be useless. Still, it's a great help having the more common macros predefined. Even if you never actually use them, just looking at a few will give you ideas on how to set up your own.

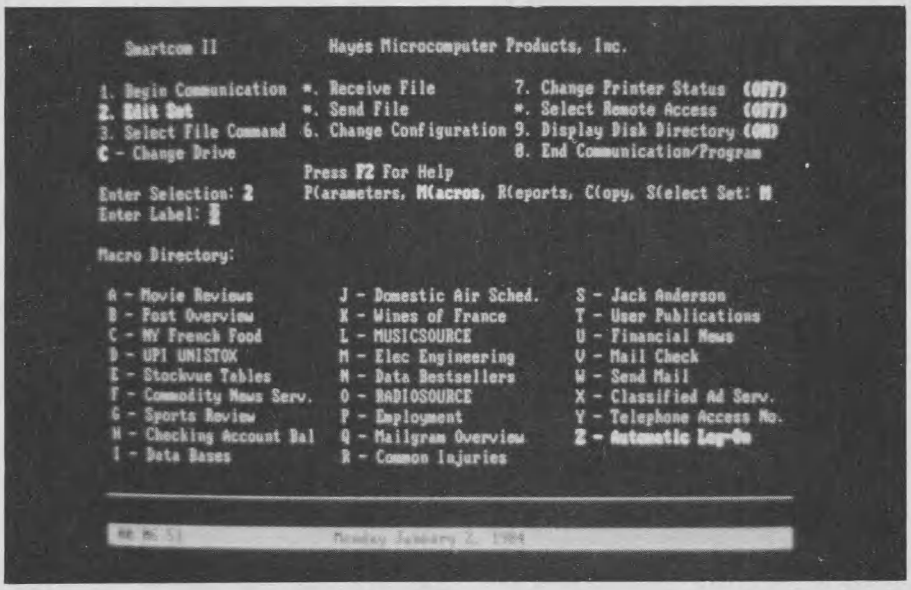


Photo 1. Smartcom II's main menu. In this screen, option 2 has just been selected. The "macro directory" in the bottom half of the screen is a listing of the preprogrammed macros for The Source.

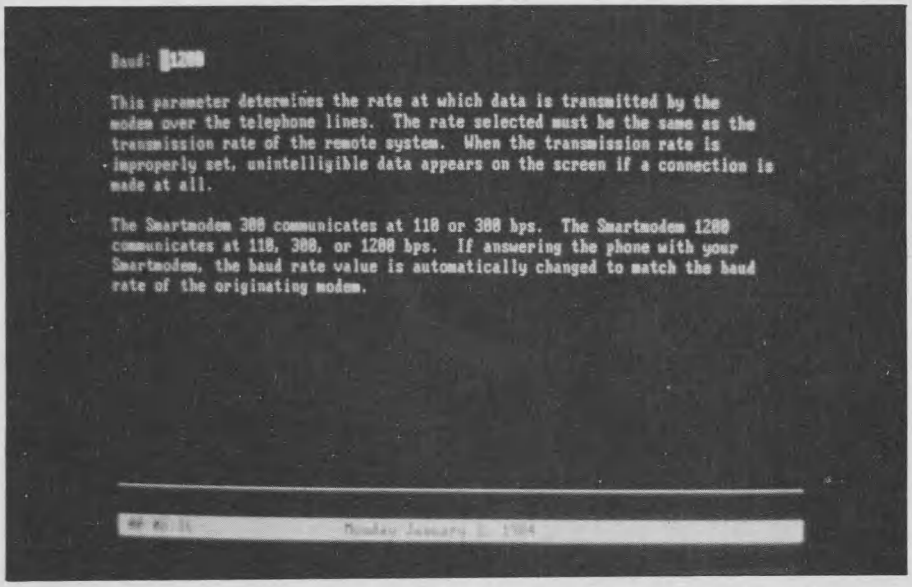


Photo 2. One of the Smartcom II's excellent help screens. Here, the help key (F2) was pressed in response to a prompt asking for a bit rate. Notice that the prompt remains on the screen when the help message is displayed.

Unique Features

Each program has some features that the other lacks. Perfect Link has a very valuable feature that lets you read and write to some CP/M format disks, provided that you have a two-drive system. The program currently reads and writes the following disk formats: Epson QX-10, Kaypro II, 4 or 10 (double density), NEC 8001, CP/M-86 (single or double density), Osborne I and Executive (double density) and Zenith Z-90 737.

Perfect Link also has an install program that makes it easy to set up a communications set for any of six information services (The Source, CompuServe, Dow Jones, NewsNet, Knowledge Index, Official Airline Guide). When you run the install program, it asks you for your area code, the type of modem you're using and the information service you want to install. Next, it displays a list of cities in or near your area code that have switching networks and asks you to select the one closest to you.

After you choose a city, it asks if a call to that city is a toll call. Supply the program with your ID number and password and you're ready to use that particular information service. After you've installed one service, the program remembers your area code and modem type, eliminating the need to supply that information the next time you install a service.

Perfect Link's terminal emulation features are better than Smartcom II's. Perfect Link can emulate a Televideo 920, a VT-52, an ADM-3A, an IBM 3101, a Teleray or simply a dumb terminal. Smartcom II supports standard control codes for carriage return, line feed, backspace, tab, bell and clear screen as well as direct cursor positioning using an escape sequence.

Smartcom II's unique features include a large display buffer (its size depends on how much memory your computer has) that you can scroll through using the cursor-movement keys. It's great to be able to scroll back six or seven screens to read something over again.

Smartcom II has another feature called remote access that lets you transfer files to and from an unattended computer that's also running Smartcom II. The remote access feature also lets you assign a password to the remote system, which keeps unauthorized callers from your data.

While in remote access, you can change the remote system's logged drive, display its directory and send, receive and erase its files. Smartcom II even makes adjustments when the two systems are running different disk operating systems, meaning that a CP/M machine and an MS DOS machine are considered similar by the program.

The most important feature Smartcom II has that Perfect Link lacks is on-line help. Smartcom II's help is outstanding. Pressing F2 in response to any program prompt displays a clear message telling you what you're doing and what the program is expecting. The help feature is context-sensitive, meaning that the program knows exactly which prompt you were faced with when you asked for help and displays a message pertinent to that prompt.

Documentation

Although both programs are well-documented, I prefer Smartcom II's manual. Perfect Link's is about 300 pages long, perfect bound (appropriately enough!) and attractively designed. It's written in a light, nontechnical style; in fact, the introduction states: "Each section includes just enough information to let you attain your goals." Its first chapter is designed to get you telecommunicating immediately.

Other chapters give specific information on accessing all the popular information services—how to subscribe, what each offers and a sample session. The manual is loaded with screen diagrams and illustrations, and it has a decent glossary and a superb index. My only complaints are that its style is a little self-congratulatory (it keeps drilling into you how good Perfect Link is) and that its binding prevents you from laying it flat next to the computer.

Smartcom II's manual is different in almost every way. It's about 210 pages long, beautifully designed and illustrated and bound in an 8½ by 9-inch, three-ring binder with tabs separating each section. It's written in a clear but dry style and is very thorough. It contains a 20-page communications primer that gives you a firm background in data communications and includes chapters on the major information services.

The manual has a 13-page troubleshooting section (Perfect Link's is only 4 pages long), numerous checklists to get you through complex tasks and lots of screen diagrams. It has a detailed table of contents but, unfortunately, no index. Its loose-leaf binding and tab separators make it far easier to use than Perfect Link's, especially when you're frantically looking for something during a communications session.

One nice bonus that comes with both packages is a set of special offers from some of the biggest information services. Perfect Link comes with coupons "worth up to $400," while Smartcom II comes with a sheet describing discount offers from Dow Jones and The Source.

Two Roads Diverged

While both programs are powerful, menu-driven and easy to use, each takes a different approach to menus and option selection and each looks and operates differently.

When you run Smartcom II, the first thing you see (aside from a brief title screen) is a 10-option menu (Photo 1). You can select an option by either

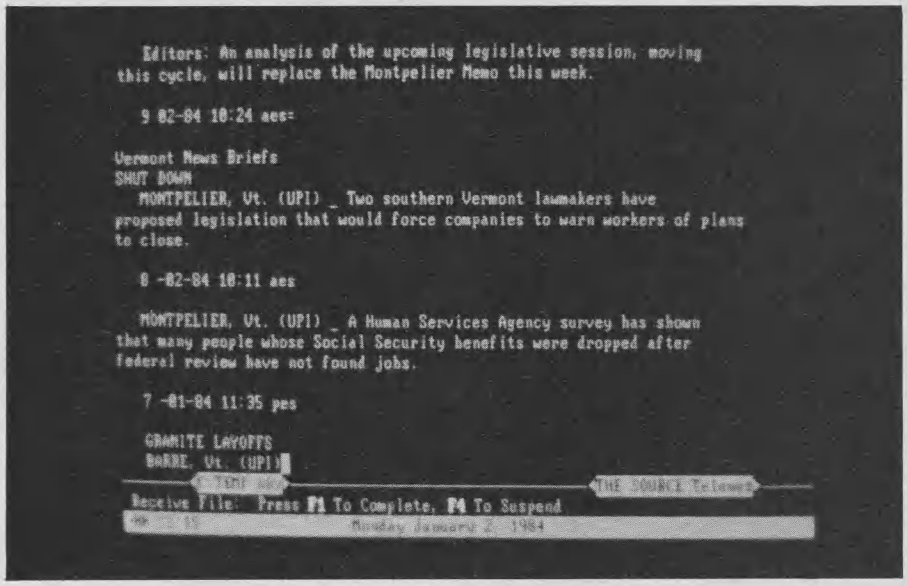


Photo 3. On-line with Smartcom II. Here, information is being captured on disk as it's being received.

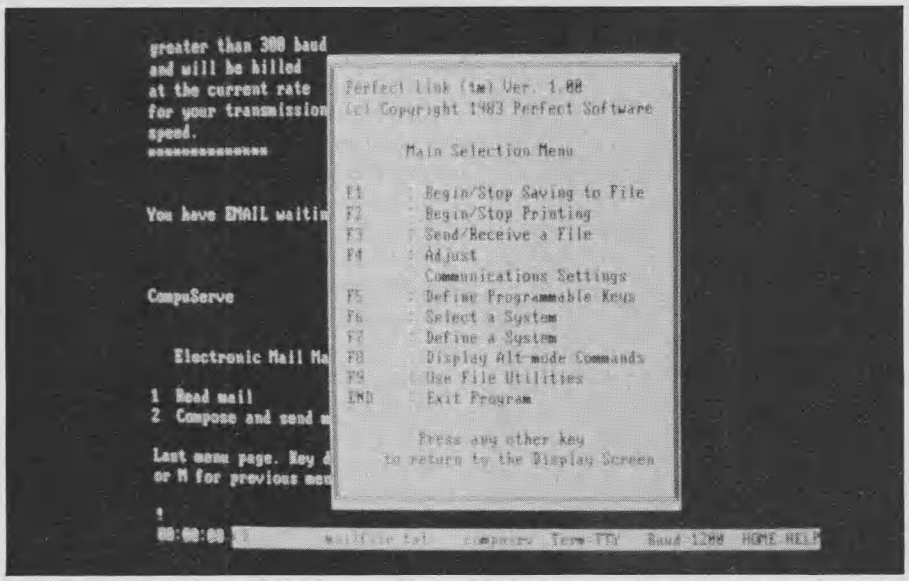


Photo 4. On-line with Perfect Link. The status line at the bottom of the screen displays (from left to right) the amount of time on-line, whether or not a carrier has been detected (CD), the name of the file that you're saving to (if any), the name of the communications set in use, which terminal Perfect Link is emulating, the bit rate and a reminder that you can display the main menu by pressing the Home key. In this photo, the Home key has just been pressed, displaying the main menu.

typing its number or by pressing the left and right arrow keys until the option you want is highlighted and then pressing return. Below the menu, a descriptive sentence tells you more about the currently highlighted option. All menus in Smartcom II work this way.

Two status lines are at the bottom of the screen the entire time the program is running, providing different information at different times. Sometimes they remind you of which function key to press to perform a given task; other times they display error messages; still other times they tell you what the Smartmodem is doing. At all times they display the time and date. Smartcom II's screens are more than functional—they're attractive and legible at all times.

Perfect Link is more awkward to use. It carries the menu concept and the use of function keys to an excruciating extreme. Every menu requires that you press function keys to select options; therefore, no menu can have more than ten options (that's why you're limited to ten preprogrammed communication sets and 20 programmable macro keys).

This restriction forces you to go through a maze of sub- and sub-submenus to select certain options, especially certain protocols. The program's full menu tree, printed in the manual, looks like a 100-year-old maple.

Perfect Link has only one status line and, because its messages are generally short two-letter abbreviations, it's less useful than Smartcom II's. The program lacks on-line help. The one help message it does have displays its main menu. That's right: the main menu is not displayed unless you ask to see it.

In all fairness, once you memorize which function keys do what, you can get through the program quickly. I still prefer Smartcom II's approach; it requires less memorization and is not so menu-maniacal.

And Now, the Winner

Both Perfect Link and Smartcom II are powerful programs. Both seem bug-free, come with good instructions and are from well-established firms that offer support. Either one should take care of most peoples' telecommunicating needs.

Perfect Link is fairly easy to use once you get used to its menu style. Its ability to read other disk formats is a valuable feature that Smartcom II lacks, and its install program makes getting started a cinch.

Smartcom II, however, is better designed, has excellent on-line help, a powerful remote access feature, a display buffer and more programmable macro keys and bigger macros.

My conclusion? Perfect Link is pretty good; Smartcom II is outstanding.■

A Capsule Look At Smartcom II and Perfect Link

	Smartcom II	Perfect Link
Product name and manufacturer:		
	Hayes Microcomputer Products, Inc.	Perfect Link
	5923 Peachtree Industrial Blvd.	Perfect Software, Inc.
	Norcross, GA 30092	702 Harrison St.
	$149	Berkeley, CA 94710
	$599 with Smartmodem 1200B	$149
System Requirements:		
	IBM PC or compatible with color or monochrome display (monochrome recommended)	IBM PC or compatible
	96Kb memory	128Kb memory
	1 disk drive	1 disk drive
	serial communications card (free-standing modem only)	serial communications card (free-standing modem only)
	suitable 300 or 1200 bps modem with appropriate cables	suitable 300 or 1200 bps modem with appropriate cables
	MS DOS 1.10 or 1.00 (see text)	MS DOS 1.10 or 1.00 (see text)

A Plug-In Smartmodem for the IBM PC

The modem I used in my review of Perfect Link and Smartcom II was the new Hayes Smartmodem 1200B. The 1200B sells for $599, including Smartcom II; the modem is not available without the software.

The Smartmodem 1200B is functionally equivalent to the freestanding Smartmodem 1200 connected to a serial communications card. The 1200B, however, fits in one of the PC's expansion slots, eliminating the need for the serial adapter and freeing up space on your desk.

There are some other differences between the 1200B and the freestanding 1200. The 1200B has only three configuration switches against the 1200's eight. This means that certain 1200B functions can be controlled through the UART registers only. The 1200B supports software modem reset; with the 1200, you often have to shut off the modem to reset it.

The 1200B adds a jack for your telephone, while the 1200 requires a Y adapter to attach a phone to the same line. The 1200B interprets the I (identification) command differently than the 1200. The I command simply returns information about the modem—its type, version number and a ROM checksum. Finally, since the 1200B is an internal modem, it lacks the 1200's LEDs.

The 1200B uses the same command language and result codes as the 1200. Both modems use the same microprocessor and firmware.

Installation

The 1200B comes with information that makes installation easy. The instructions are clear and illustrated with photographs. To install the modem, you simply remove your PC's system unit cover, select an unused slot (you do have an unused slot, don't you?), remove the slot cover, snap a plastic card guide into place and plug in the board.

Then simply screw a retaining bracket into place, replace the system unit cover, attach the modular cable to the modem and the phone lines and, if desired, plug a telephone into the modem. The whole process should take a rank beginner no more than 20 minutes—that includes attaching the FCC sticker to the back of the computer (you will attach the sticker to the back of your machine, won't you?).

Manual Matters

The 1200B comes with a spiral-bound, 5 × 7½-inch, 75-page manual. Most users will be interested in only the installation chapter. Other chapters provide technical information on the modem's commands, registers and error messages and describe how the modem interfaces with the PC's expansion bus. There's also an interesting appendix describing how the modem can be used in amateur radio applications. A pull-out reference card of Smartmodem commands is also included.

Elegance

What can I say? I tried my Smartmodem 1200B in an IBM PC and in a Compaq Plus, and it performed flawlessly. Combined with Smartcom II, it makes an elegant communications system and one of the best accessories you can add to your machine, especially if you have a transportable PC clone.

J.H.

A Peek at Statpro

This preview of Statpro for the IBM PC should give you an idea of what you can expect when this powerful statistical program is available.

By Shawn W. Bryan

Wadsworth Professional Software has announced the release of the IBM PC version of its statistical program, Statpro.

The program, written in Pascal and operating under the p-system environment, includes many statistical features of larger mainframe packages (like SPSS) with the ease of use and cost effectiveness of a microcomputer.

Statpro is designed for the person who makes his living crunching numbers. It isn't designed to teach you about statistics, nor is it a simple business forecasting package. For the price ($1995), you would expect this to be a sophisticated package, and it is.

Descriptive	Crosstabs and contigency
	Comparative
	Stem and Leaf counts
	Chi-Square
	Descriptive
	Normality testing for residuals
	Non-parametric Range Stats
	Non-parametric comparisons
	Letter value displays
Regression Analysis	Linear
	Nonlinear
	Statistical matrices
	Multiple regression
	Stepwise multiple regression
	Residual analysis of stepwise and multiple regression
ANOVA	Single classification
	Nested classification
	Two and three-way equal classification
	Kruskal-Wallis Classification
	Two and three-way unequal classification
Time Series Analysis	Moving averages
	Multistage lease squares
	Polynomial and trig functions
	Exponential forecasting
	Forecasting additive series
	Forecasting multiplicative series
Multivariate Analysis	Principal components
	Principal axes
	Canonical correlation
	Matrix inverse and determinant
	Pair weighted cluster analysis
	Multiple contingency analysis
	Discriminant function analysis

Table 1. A list of Statpro's statistical routines.

The program should be on store shelves by the time you read this review. The graphics modules for the program are scheduled for release in March 1984. A prerelease version of the program was made available to me for this preview.

When a final version of Statpro is available, I hope to complete a comprehensive review of the program's capabilities. Until then I'll tease you with a quick look at what it is supposed to do when completed.

An Eight-Pack?

The program comes on eight separate disks. The major statistical programs and the database management module are located on separate disks. In addition to the program module, each disk contains the p system. The way Statpro is implemented, each disk stands independent of the others. The people at Wadsworth used the NCI implementation of the p system and Pascal. This is the fastest implementation of this operating system around (that I'm aware of). It also uses ten-sector formatting of the disks, providing additional storage. Since the program is cramped for space, the ten sectors are a necessity.

For more information on Statpro, contact Wadsworth Electronic Publishing Co., Statler Office Building, 20 Park Plaza, Boston, MA 02116.

Statpro's database manager is normally your first stop in this program. It is here that data entry takes place. The process is simple and straightforward, but is slowed by the speed of the screen refresh. I can touch-type numbers faster than the program accepts them. Other than that, the data entry part of the program is excellent. There is room provided for up to 72 data fields. Fields can be flagged, searched and sorted, and data can be transferred between fields. Records can be processed individually or in batch mode. Misplaced data should be a thing of the past with this program.

Data transformation and conversion is also accomplished in the file manager. Almost every conceivable arithmetic transformation is provided for. If you don't find the one you want, you can create your own formula to do transformations. Fields of normal deviates, proportions and center of range can also be created. Finally, Box-Cox transformations and polar/rectangular conversions are possible. A conversion library is also provided to let you convert fields from one kind of data to another (miles to kilometers, for example).

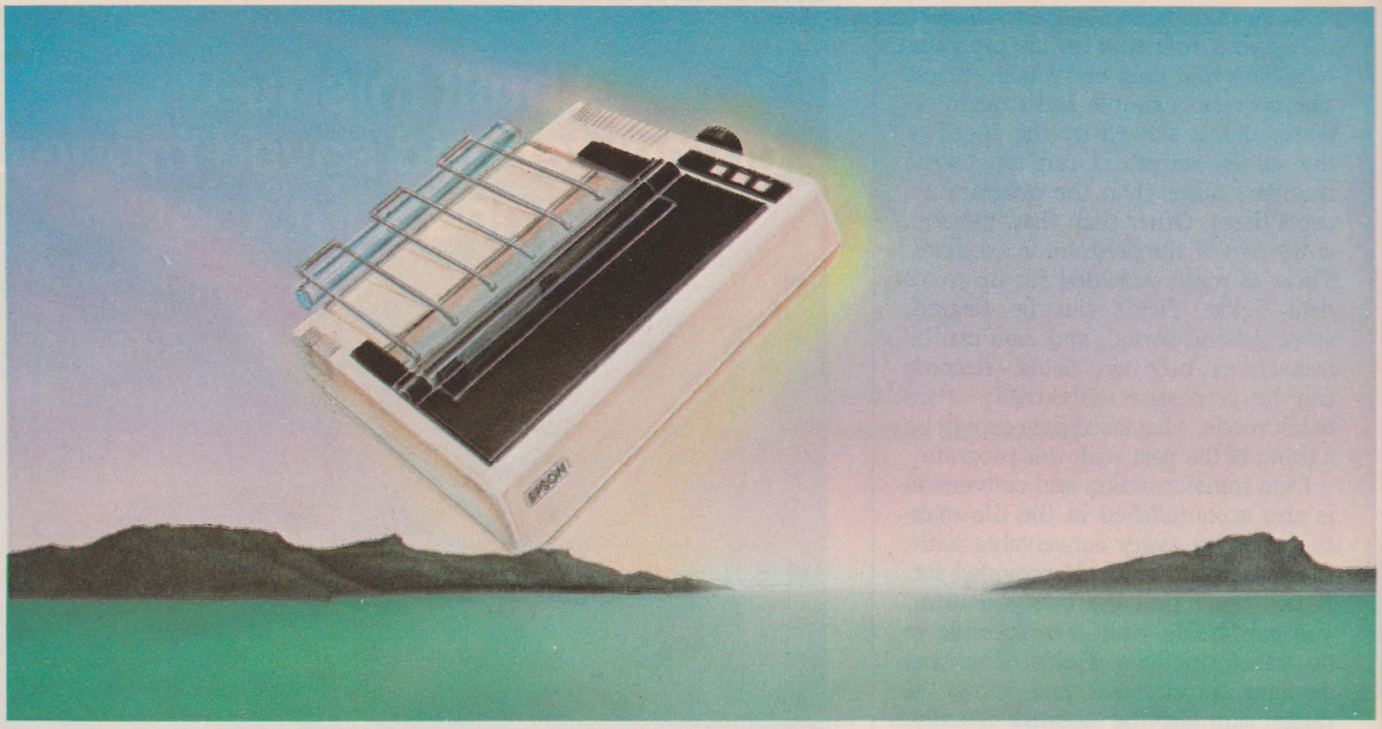
While I haven't tested all of its features yet, Statpro offers an impressive list of statistical routines (Table 1).

It's apparent that this package has a great deal of depth and that a working statistician will probably find it covers 90 percent of the problems he wishes to analyze.

Graphics functions aren't available yet, but will be reviewed in depth with the final marketed product. It's interesting to note that the advertisements claim that Statpro will provide for most of the conceivable types of graphs. The graphs will be backed up with an impressive list of screen handling functions including multiple plots, intermixed text and graphics, and, if desired, joystick control of editing.

The report formatting of Statpro is exemplary. The reports created by the program are of presentation quality and a number of popular printers are supported.

If you're considering purchasing a stats package, you may want to wait for this one to actually hit the market. Support from the company has been good. I called with several questions and was quickly and ably assisted. While I can't recommend a product that isn't ready for market, I can suggest that this one may be worth waiting for.■



Is There an Epson

The author had an Epson MX-80 printer and a North Star Horizon Computer. He knew the MX-80's parallel port could be connected to Horizon's standard serial port, but how complex would the interfacing be? It turned out to be less difficult than expected, and it costs less than $20.

By John Uffenbeck

On Your Horizon?

Finally, I had had enough. Although I had an Epson MX-80 printer sitting on my bench, I was using an old ASR-33 Teletype to obtain copies of my North Star Horizon programs. Connected to the Horizon's standard serial port, the interface was simple. But at 110 bits per second and uppercase only, it wasn't exactly the ideal word processing system.

I knew the Horizon and MX-80 could be interfaced, but the complexity of a parallel port with its strobe lines and handshaking signals had scared me off.

This time, though, I decided to solve the problem. After all, someone who teaches electronics for a living ought to be able to connect a few parallel wires in the right places.

As usual, the project turned out to be simpler than I thought. The total cost was less than $20 (assuming you have a parallel port on your Horizon), and the software required changing only two bytes in the North Star's standard parallel output routine.

You Want Hard Copy?

If you have a North Star Horizon computer and have been wanting to add "hard copy" to your system, read on. I'll describe how I interfaced an Epson MX-80 printer to the Horizon computer and modified North Star's DOS to make it all work.

Before we begin, a few comments about the two common types of printer interfaces are in order.

Serial and Parallel Interfaces

Printer interfaces can be serial or parallel. A serial interface can be simple, requiring as few as two connections (data and ground) between the

---

John Uffenbeck (Rt. 2, Box 56, Cable, WI 54821), the author of Hardware Interfacing with the TRS-80 *(Prentice-Hall, Inc., 1982) is a professor of electrical engineering at the University of Wisconsin-Platteville.*

---

computer and printer. Data is transmitted one bit at a time down the data line as shown in Fig. 1.

Because all computers are internally parallel machines, some means of converting their data to a serial format is required. This is usually done with a universal asynchronous receiver-transmitter (UART) chip.

In addition, to follow RS-232C standards, the logic levels must be typically −12 V for a logic 1 and +12 V for a logic 0. Because this is incompatible with standard TTL (transistor-transistor-logic), special line drivers and receivers are also required.

The advantage of having an RS-232C interface is standardization. If your computer has an RS-232C port and your printer has an RS-232C input, you need only a two-conductor cable between the two to be "up and running." This should be true if your computer is a North Star, a TRS-80 or an Apple. It should also be true if your printer is made by Qume, Centronics or Epson.

A serial port can also be interfaced with a modem to give you access to larger computers and information networks over the telephone lines.

Serial interfaces can be a bit more complex than I'm making them sound. Often, a clear-to-send signal must be supplied by the printer to avoid a loss of data when the printer is off-line.

In addition, the protocol between the computer and printer must be matched. This includes the bit rate, parity bit selection and the number of data bits and stop bits per data word. Usually, DIP switches are provided on the printer to allow setting these parameters.

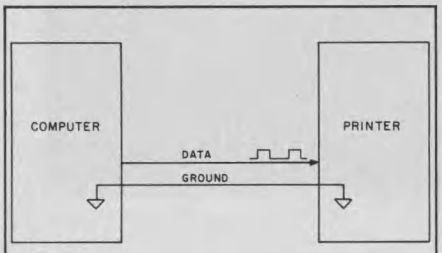


Fig. 1. A serial printer interface can be simple, requiring as few as two wires. The serial data is transmitted bit-by-bit down the Data line.

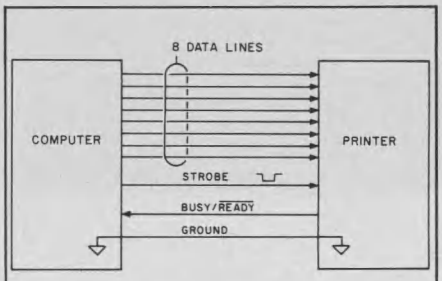


Fig. 2. A parallel printer interface requires 11 wires—eight for the data bits, two handshaking signals and the ground wire.

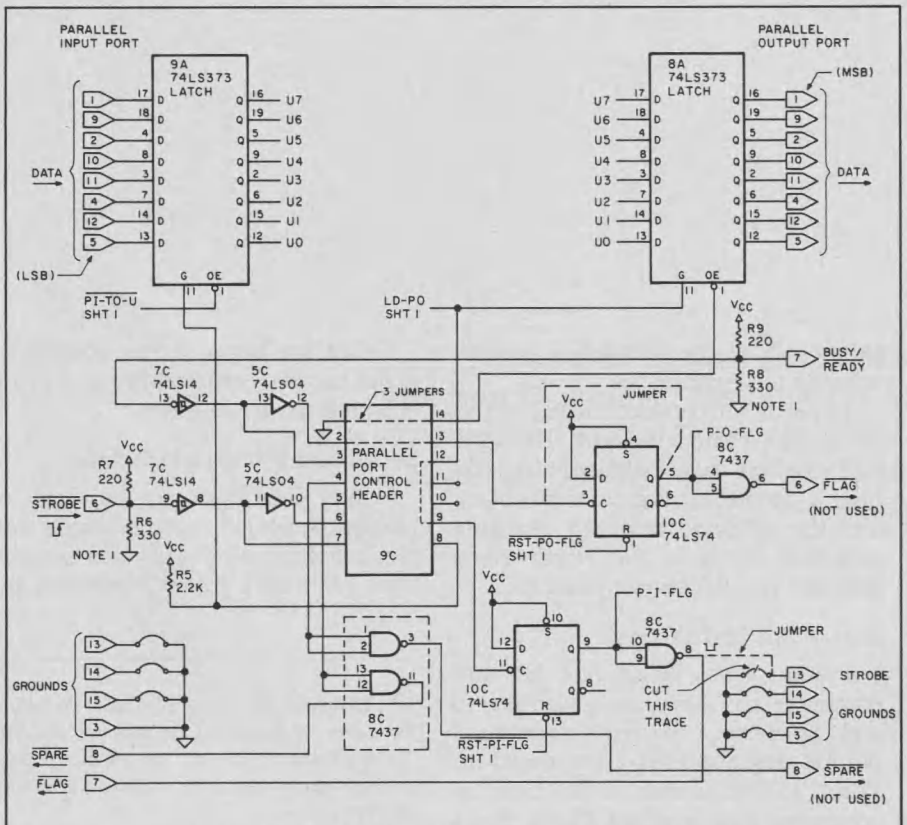


Fig. 3. Parallel printer interface between the North Star Horizon and the Epson-80 printer. The circuit shown is the standard Horizon parallel printer output port, but it's modified to accept the BUSY/READY status of the printer and to provide a data strobe pulse.

Be Careful . . .

A parallel printer interface is shown in Fig. 2. The computer supplies eight bits of information that represent the character to be printed. But now we must be careful.

The Z80 microprocessor running at 4 MHz is capable of outputting a new character every 3–5μs. That's more than 200,000 characters per second! Maybe IBM's latest ink jet printer can handle this, but the MX-80 can't.

Because of its relatively slow speed, the printer must supply a synchronization signal, usually called BUSY/READY. When this line is high, the printer is busy (printing a character) and cannot accept a new character.

So here's where the mix of hardware and software comes in. A simple machine language routine is used to monitor the BUSY/READY flag waiting for it to go low; then and only then should a new character be output to the printer.

To tell the printer a new character is available, the computer usually supplies a brief pulse or strobe signal. As Fig. 2 illustrates, this requires a total of 11 wires (compared to a serial port with as few as two wires).

The parallel interface thus requires an eight-bit output port, a one-bit input port and software to monitor the BUSY/READY status line. When BUSY/READY is low, new data should be output and a strobe pulse generated.

Most parallel printers also provide Paper Out, Unit Select and Error signals. However, because each of these also causes BUSY/READY to be set, it's not necessary to monitor these flags.

Modifying the Horizon

When equipped with the optional parallel I/O interface, the Horizon computer provides separate eight-bit parallel input and output ports. North

Star uses a header plug to determine the exact configuration of each port. Fig. 3 illustrates the parallel printer interface and the changes that must be made to accommodate the Epson printer.

First, remove the motherboard from the Horizon chassis noting the proper connection point for each lead coming from the power supply. The following three steps are required to modify the motherboard for the printer interface:

1. Configure the parallel port control header plug so that pin 1 is connected to pin 14, pin 2 is connected to pin 13 and pin 9 is connected to pin 12. These connections permanently enable the tri-state output latch, route the printer's BUSY/$\overline{\text{READY}}$ signal to pin 3 of IC 10C and cause the LD-PO pulse to set the P-I flag. This will occur with each I/O write to the output latch. The inverted P-I flag will then serve as the printer strobe pulse.

2. Bend pins 3 and 5 of IC 10C (74LS74) so that they no longer fit into the socket. On the bottom of the motherboard, tack-solder a jumper wire between these two pins. With this modification, the PO flag will follow the BUSY/$\overline{\text{READY}}$ status of the printer. Note: This modification prevents BUSY/$\overline{\text{READY}}$ from clocking IC 10C. This is necessary to prevent a loss of data when an error condition occurs (BUSY/$\overline{\text{READY}}$ changes to a 1 but the flip-flop still shows READY).
READY).

3. On the bottom of the motherboard, tack-solder a jumper wire from pin 8 of IC 8C (7437) to pin 13 of the parallel output connector. On the component side of the board, cut the narrow piece of foil that connects pin 13 to ground.

This concludes the modifications to the motherboard; you may replace it in the Horizon chassis. Take care to connect all of the leads from the power supply to their proper connections on the motherboard.

Wiring the Connectors

This is the tedious part of the job. I ordered a 36-pin male connector (Amphenol 57-30360) from Orange Micro, Inc. (3150 E. La Palma, Suite G, Anaheim, CA 92806). The total cost was $11.48, including shipping, and the connector arrived only four days after I placed my phone-in order.

Next, I purchased a five-foot, 40-conductor ribbon cable from Radio Shack. The cost was $6.95. Soldering the 36 wires to the printer and computer connectors is not an enjoyable task; you probably should allow 2-3 hours for the complete cable assembly.

I started by pulling off the four unused conductors from the cable. Now you can do one of two things. You can solder all 36 connections to the printer connector or press fit the cable and connector. The last choice is easiest but requires care. The center of each conductor must line up with the groove in each pin. Holding the cable in place, replace the connector cover and carefully squeeze with a vice.

The computer end of the cable can now be prepared. You should have received a 15-pin male connector with the Horizon parallel port.

The job now is to connect the appropriate wires coming from the printer connector to the corresponding pins on the computer connector. Table 1 indicates the pin matchups. I suggest using an ohmmeter to be sure you have the proper wire before soldering it in place.

The Epson manual suggests that every signal wire be surrounded by a ground wire to reduce noise pickup. This accounts for the large number of ground wires in Table 1. Because only pins 3, 14 and 15 are ground at the computer connector, you'll have to terminate all of the ground wires at these three pins. I soldered a piece of resistor lead between the three pins, making this job somewhat easier.

Finally, when you're finished (and you appreciate why a "simple" printer cable can cost $35), test each pin for proper continuity with your ohmmeter.

Software

North Star DOS already has a parallel output routine; you need to change only two bytes to make it compatible with the interface described here. An

North Star (Computer Connector)	Signal Name	Epson MX-80 (Printer Connector)
pin 5	data 1	pin 2
12	2	3
4	3	4
11	4	5
10	5	6
2	6	7
9	7	8
1	8	9
13	STROBE	1
7	BUSY/READY	11
3,14,15	GROUND	16,17,19-30 33,36

Table 1. Pin number matchups for the MX-80 and the Horizon parallel output port.

```
2954                *SAMPLE PARALLEL OUTPUT ROUTINE
2954                *COPIED FROM NORTH STAR MANUAL
2954 DB06     COUT2 IN    6       READ MOTHERBOARD STATUS
2956 E601           ANI   1       MASK FOR BUSY/RADY FLAG
2958 CA5429         JNZ   COUT2   PRINTER NOT YET READY
295B 78             MOV   A,B     GET CHARACTER TO ACC
295C D300           OUT   0       OUTPUT DATA TO PRINTER
295E 3E30           MVI   A,30H   LOAD COMMAND BYTE
2960 D306           OUT   6       TERMINATE STROBE PULSE
2962 78             MOV   A,B     GET CHARACTER TO ACC
2963 C9             RET
```

Listing 1. Parallell output routine for the MX-80 interface.

8080 machine language program is provided in Listing 1.

Two changes are made to the routine given in the North Star System Software Manual (I'm using release 4 DOS):

- The JZ at location 2958H is changed to JNZ (wait for BUSY/READY to be low).
- The byte at location 295FH is changed to 30H (this will reset the 74LS74 flip-flop and terminate the strobe pulse).

The MX-80 requires an active low strobe pulse of 0.5μs or longer, but it may not be clear to you as to how the hardware accomplishes this. The OUT 0 instruction sends data to the 74LS74 latch. And, because pins 9 and 12 of the control header plug are wired together, OUT 0 also sets the 74LS74 flip-flop. The 7437 inverts this level to produce a logic 0 on pin 13 of the computer connector (STROBE).

STROBE will stay low for the time it takes the Z80 to execute the MVI A,30H and OUT 6 instructions. This is 18 clock cycles, or 4.5μs with a 4MHz system clock. In this way, a 4.5μs strobe pulse is generated "automatically" without the need for a one-shot. Fig. 4 illustrates the timing.

Listing 2 details the procedure to be used to patch the new parallel output routine permanently into your DOS. Load the disk to be modified into drive 1 and follow the steps shown. When completed, reboot the system and type LI #2. The catalog entry of your disk should be typed out on the MX-80. Be sure the printer is in the on-line position.

From Basic, the following commands can be used with the printer:

LIST #2	List the current program.
PRINT #2 "string"	Print the "string" data.

Conclusion

In this article, I have described a parallel printer interface that allows the Epson MX-80 printer to be used with the North Star Horizon computer. In fact, once your Horizon has been modified, it should be compatible with any parallel printer. You'll have to rewire the printer end of the cable, however, to match the pin designations of the printer used.■

```
+LF  DOS 4000              load DOS into RAM
+GO M2D00                  load and run the monitor
>MONITOR 5.0               monitor responds
>DS 4858                   display the location of the JZ
4858: CA=                  this is JZ
C2                         change to JNZ
>DS 485F                   display location of the 20H byte
485F: 20=                  this is the 20
30                         change to 30
>OS                        return to DOS
+SF DOS 4000               save the mofified DOS
```

Listing 2. Follow this procedure to patch your DOS with the new parallel output routine.

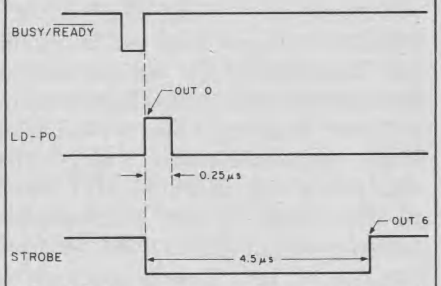


Fig. 4. Timing sequence to print one character. When BUSY/READY drops low, an OUT0 instruction is executed, loading one character into the output latch with the LD-PO pulse. This pulse also sets IC 10C, producing the STROBE pulse until the OUT 6 instruction is executed (4.5μs later).

THE INTERNATIONAL VIDEO GAME OF THE YEAR COMPETITION HERALDS THE "NEW AGE" IN VIDEO/COMPUTER GAMES

$175,000 TO BE WON

International Video Game of the Year

Create a brilliant, new video game and you could be on your way to becoming a millionaire. This fantastic competition, organised by I.R.P. (The International Register of Independent Computer Programmers Ltd) and the famous Mark McCormack International Management Group, offers programmers and inventors the opportunity of a life time. There are huge, immediate cash prizes and the on-going revenue of 10% of the sales of all games to distributors throughout the world, plus the chance to appear on an international TV show. Your skill and imagination could bring you fame and fortune!

$100,000 FIRST PRIZE! PLUS FIVE $15,000 RUNNER-UP PRIZES!

Devise a totally original new video game in one of these categories: SPORTS, SIMULATORS, ARCADE, STRATEGY, ADVENTURE/FANTASY or a special section which covers programmes that are not necessarily games but have outstanding Educational or Entertainment merit. We'll also be announcing a number of 'MERIT' awards which will be entitled to carry the message 'An International Video Game of the Year MERIT AWARD' on their retail packaging. It's a great challenge. And the rewards, both financially and in terms of prestige, are tremendous. This is the most exciting competition ever for creative computer and video enthusiasts.

YOU'RE A TV STAR TOO!

All six winning games and their inventors will be featured on an internationally distributed, spectacular TV special. That's going to make your name!

HOW TO ENTER

Just send in your game, or games, programmed on cassette for any popular home computer. Use the coupon, today, and we'll send you all the facts you need.

CLOSING DATE FOR ENTRIES IS 31st MAY 1984

To: **IRP Limited, Pinewood Film Studios, Iver, Bucks, England.**

Name .....

Address .....

.....

.....

M1

.....

Hot Rod Your Kaypro

For a $50 investment, this ROM chip and a few modifications can turn your Kaypro into a souped-up, two-speed computing monster.

By Thomas Howe

A lot of fine folks have been devising ways to make your Kaypro computer run faster and more efficiently. They are dispelling the myth that spread when the machine was first introduced—that it couldn't be expanded. Indeed, the Kaypro can be ingeniously hot-rodded into a super machine.

Not the least of these geniuses are the people at Micro Cornucopia of Bend, OR. They devised a simple and inexpensive way to make the Kaypro run at either of two speeds (the stock 2.5 MHz and the faster 5 MHz) using easily obtained parts and about an hour's work. The nicest thing is that you can do it for less than $50.

The faster speed really enhances Perfect Writer, The Word Plus and Perfect Formatter, to name just a few. You will also find database applications that do a lot of sorting will perform better at the faster speed. Programs that used to crawl now take off and fly. PerfectCalc no longer creeps.

This project will allow your Kaypro to run at two speeds. Why can't you just leave your Kaypro at 5 MHz all the time? Some software doesn't tolerate the high speed, so you must be able to retain the slower speed for which the machine was built.

This modification also works fairly well with Kaypro IIs or 4s that have the 16-bit Plus 88 board installed. The RAM disk feature of the Plus 88 models performs nicely at 5 MHz. There are some problems with MS DOS that do require slow speed operation when using IBM-type software.

Under the Cover

The procedure is actually simple and enjoyable. The folks at Micro C have done all the hard stuff, like figuring out what makes the Kaypro tick. All you have to do is learn some computer anatomy and be able to identify a few parts. Then you change two chips and solder six connections.

There is one thing that you should remember about any project like this. In the process of altering the computer, you void your warranty. You should wait to hot rod your Kaypro until it is out of warranty, because you'll be on your own after you do it.

Having been warned, it's now time to take the lid off your Kaypro and look around inside. You only have to remove ten Phillips-head screws with a #2 screwdriver. Be sure to pull the power plug, and don't touch any of

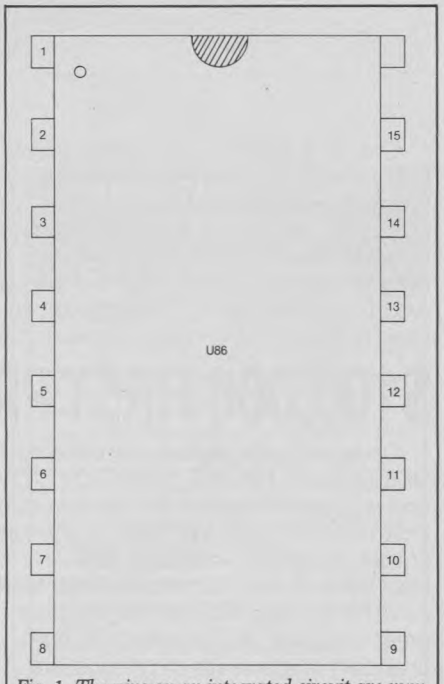


Fig. 1. The pins on an integrated circuit are numbered counterclockwise. Note that the chip should be facing away from you.

Address correspondence to Thomas Howe, 255 Flamingo Road, Mill Valley, CA 94941.

the large capacitors (I wouldn't want you to be shocked).

Find the main board at the top of the computer, above the CRT. Looking at it from the front of your machine, find the CPU, the nerve center of the whole machine. It is near the front center of the board (see Photo 1). It's a standard Z80 microprocessor that retails for about $4 or less.

Chips to Left, Chips to Right

To the left of the CPU are eight chips, labeled U20 through U28. These are the memory chips, offering a total of 64Kb of RAM.

The back left quarter of the board controls your video, handling communications between the computer and the screen.

To the right of this, in the back right quarter, are the communications chips. These handle the computer's conversation with the outside world. They drive your printer, service your modem and, on the Kaypro, talk to the keyboard.

In the front right corner of the main board are the disk-controller chips. Without this section, you would never be able to use the disk drives.

These key elements are all housed on the Kaypro's 8½ × 11½-inch board with room to spare. Many single-board computers are much larger and have many more chips. Kaypro has overhauled the basic design and economized it with simplicity. Most chips are newer and more powerful versions of what was available when the original big boards were conceived.

An Anatomy Lesson

The stock Kaypro already has the necessary system clocks for running faster; they are simply blocked out for reasons unknown. I should also point out that the memory chips are capable of running faster, so there will be no problem with the extra speed you're about to give the machine.

If you look carefully on the front center edge, you'll find a crystal (the shiny, flattened cylinder with 20.000 written on it) on top of a little foam pad. This crystal, along with a little circuitry, establishes the rhythm for the computer. The crystal vibrates rapidly and rest of the machine uses that regular vibration like a tuning fork to keep time.

A couple of chips are also involved. They are near the crystal and labeled U66 and U86. In some fine way, they translate the crystal's vibrations into something useful to the computer. That crystal is vibrating 20 million times per second (the meaning of 20 MHz). These chips and a few others divide that speed a few times until the machine is safely running at an even 2.5 MHz pace.

Since this job involves rearranging the wiring to a chip, you need to know how to count pins on an IC. The pins are numbered counterclockwise, beginning at one, which is located to the left of the chip's notch. (The pins should be facing away from you.) See Fig. 1. Some ICs also have a tiny circle next to pin 1; all ICs have the notch.

The next thing you have to identify is a "trace." On the board are thin copper lines going from one chip or solder point to another. These are the traces that wire your printed circuit board.

Shopping List

You only need a few simple parts to create a two-speed shift on your Kaypro II or 4. The total price you can expect is less than $50, making this the cheapest modification you are likely to do.

First, buy a new Z80B chip. It is important to get a Z80B because it can handle speeds up to 6 MHz. Some Z80A CPUs may be able to handle 5 MHz, but this would be a risky proposition.

The new CPU should be available from your local electronics supply house or from some of the mail order outlets advertising in the computer

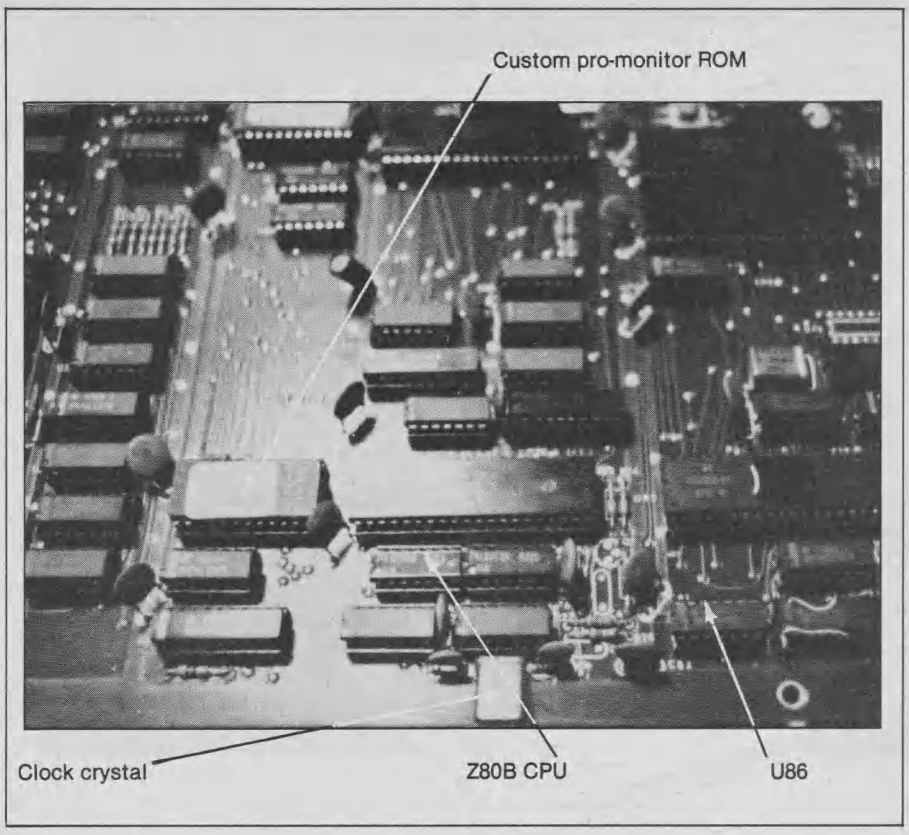


Photo 1. Here's where you locate the parts of your Kaypro board most important to this speed-up modification.

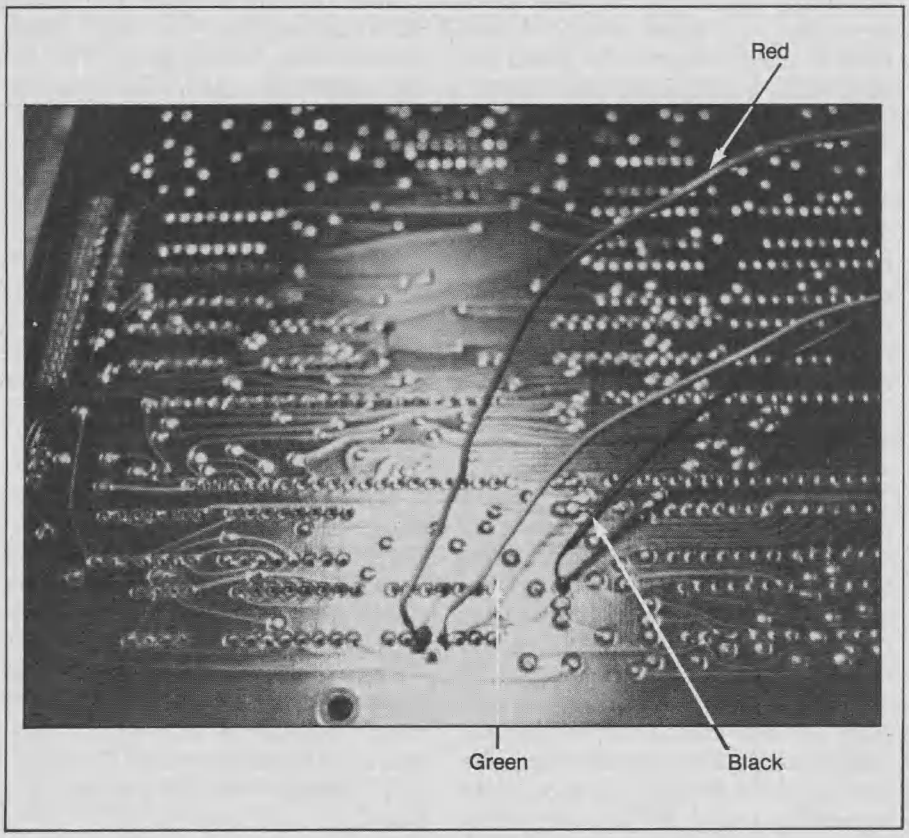


Photo 2. This view of the bottom side of the Kaypro board shows where to solder the wires under U86 and R26.

magazines. Expect to pay about $12.

Next, you'll need a high-quality, single-pole, double-throw switch. The switch should be small enough to install in one of the cooling slots at the back of the Kaypro. You can use either a slide switch or a toggle switch. Expect to pay about $3 to $5.

You also need some solid copper wire. Don't use stranded wire for this job. Radio Shack is a good source for #22 wire. They have a package containing three colors of insulation, which makes it easy later on to know which wire is which. I like red, green and black and will refer to these colors later; if you have other colors, simply replace my colors with yours.

One other item is important. You must change the monitor ROM chip (U47, immediately left of the CPU) on the older IIs. The Kaypro monitor chip is slow and not capable of running at the new, faster speed. It contains the code needed to start the machine and monitors disk accesses. The original code is quite slow.

Fortunately, Micro Cornucopia has come to the rescue. It devised a new ROM chip and is now offering it for $29.95. Its Pro-Monitor has enhanced code to give you a solid block cursor, faster disk accesses and six retries on a disk read error. I recommend you order one of these rather than try to burn your own PROM or EPROM.

Micro C has gone one step further. While they are burning the code into the new ROM chip, it is simple to add a personal touch. You can (for a modest fee) specify your own message that will show on the screen whenever you turn it on or reset it. I had my name, driver's license number and a slogan installed. This is there permanently and will always allow positive identification if the computer is ever stolen. Few thieves will have the smarts to locate this chip and change it.

You can order the Pro-Monitor from Micro Cornucopia, PO Box 223, Bend, OR 97709; the company likes credit cards. I would also advise you to subscribe to its magazine for a wonderful source of technical tips as they uncover new tricks for your Kaypro.

Getting Down to Business

The only tools you need are a #2 Phillips screwdriver, a small flat blade screwdriver and a low-wattage (15–25 watt) soldering iron. Make sure the soldering iron has a small tip. You also need a little piece of thin-gauge, electronic-grade, rosin-core solder.

BOOKS

Wayne Green Books

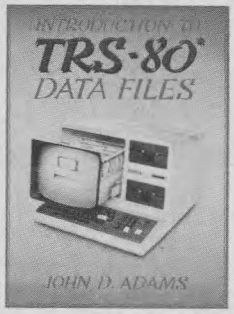
BOOKS

Rainbow Quest for the Color Computer

A computer fantasy for young Color Computer users. Rainbow Quest *is an adventure that combines fiction and programs. Readers must cross the planet Rainbow and master a series of challenges to succeed on the Quest. Each challenge is a program on cassette. Included are arcade games, puzzles, and mazes. Book and cassette sold together.* **$24.97 BK7391 128 pp.**

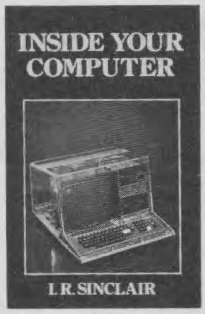
Converting to Timex/ Sinclair BASIC

Convert programs to run on your T/S 1000 or 1500. This is a guide to translating from other BASICs into Sinclair BASIC. Fifteen types of instructions are covered. The T/S replacement is given for each, followed by a description of its use and an example. Much of the material applies to the T/S 2068. **$14.95 BK7396 206 pp.**



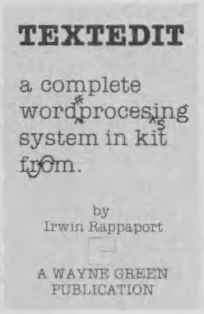
Introduction to TRS-80 Data Files

Learn by doing with this guide to writing a data base manager. This book, with its accompanying software, takes you through a simple, mailing list program to teach you about sequential and then random access files. The construction of a DBM and the techniques for moving data to and from disks are discussed. **Book and TRS-80 disk** *$24.97 BK7398 approx. 144 pp.*



Inside Your Computer

Find out what goes on inside your machine. Inside Your Computer *explains microcomputer circuits and how they work. Topics include chips, interpreters, circuits, machine language, binary numbers, algorithms, ASCII code, software, and what they all mean to the computer. Includes many photographs and schematics.* **$12.97 BK7390 108 pp.**



TEXTEDIT

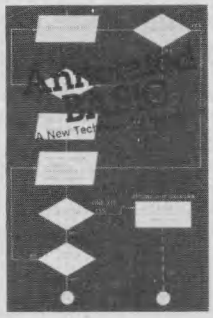
Build your own word processor with the TEXTEDIT kit. This TRS-80 Disk BASIC system is built in modules, so you can modify them or use only the parts you need. Features include complete editing, search, replace, and count, and upper/lowercase typing on an unmodified Model I. Model III users need the TRSDOS CONVERT utility to use the disk. TEXTEDIT is compatible with any major DOS. It operates with one drive; two drives or copy utility needed to transfer programs to system disk. **Book and disk package $24.97 CC7387**

Computer Carnival

For the TRS-80 Models I and III. These sixty programs for beginners will entertain and educate. Children will find mazes, word games, graphics, puzzles, and quizzes. Card games, logic tests, word and number quizzes, and letter guesses make Computer Carnival *a learning experience. The* Carnival Companion *cassette of all sixty programs is also available.* **Computer Carnival and Carnival Companion** *$24.97 CC7389* **Computer Carnival** *$16.97 BK7389 218 pp.* **Carnival Companion** *$9.97 TP7389*

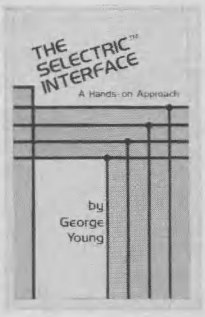
TRS-80 as a Controller

Learn to control outside devices with a TRS-80. This book is an introduction to interfacing, with simple, inexpensive projects. Applications include controlling lights and switches, building a small computer, and suggestions for more complex applications. The book applies to the Model III and, with minor conversions, the Model I. **$12.97 BK7394 192 pp.**



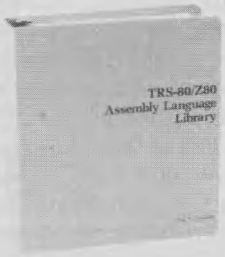
Annotated BASIC, vols. 1 and 2

This two-volume set teaches you the hows and whys of BASIC programming. TRS-80 Level II programs are taken apart and described in detail. Each program is accompanied by documentation, program annotation, BASIC concepts and definitions, and a flowchart.
Vol. 1 *$10.95 BK7384 160 pp.*
Vol. 2 *$10.95 BK7385 125 pp.*



The Selectric™ Interface

You can turn an IBM Selectric I/O writer into a letter-quality printer for your computer. The Selectric Interface *gives you the programs and step-by-step instructions you need for Selectric models 2740, 2980, and Dura 1041. With slight modifications, the instructions will work for various chips.* **$12.97 BK7388 124 pp.**



TRS-80/Z80 Assembly Language Library

Learn to use assembly language on the Model I to its full capacity. Two TRSDOS-compatible disks are included, with programs worth many times the book's cost. You'll learn about TRS-80 hardware and software, general Z80 routines, and TRS-80 utility programs. Examples show you how to apply the information to your everyday programming. Model III conversions are given. **$34.97 BK7395 355 pp. Disks included.**

40 Computer Games from Kilobaud Microcomputing

Accept the challenge of some of the best games Kilobaud *has published. This collection has everything from racing games and gambling to space adventures and calculator games for a variety of systems. Conversion instructions are given.* **$7.95 BK7381 148 pp.**

Some of the Best from Kilobaud Microcomputing

Get the programs for your PET or TRS-80 that readers chose as their favorites. Chapters include detailed programming techniques, reviews, building and interfacing, peripherals, and file structures. Applications include word processors, data base managers, and text formatters. **$10.95 BK7311 223 pp.**

Kilobaud Klassroom

Learn electronics with this hands-on course. This collection of electronics projects starts with simple concepts and takes you on to building your own small computer. You'll learn electronics theory and get the practice you need to master digital electronics. **$14.95 BK7386 393 pp.**

For credit card orders, call **toll-free, 1-800-258-5473.** Or send your order on a separate piece of paper to: **Wayne Green Books, Retail Sales, Peterborough, NH 03458.** Be sure to include the book title, order number, and price. Postage and handling is $1.50 for the first book, $1.00 for each additional book. Foreign air mail is $10.00 per book. Check, money order, or complete credit card information must accompany order. If you have questions about your order, write customer service at the above address.

Unplug and open your computer. Then carefully unplug the wires leading to the main board. You may want to mark them with a magic marker so you can later plug them back into the right spots.

Remember to be careful as you handle the circuit board. It's delicate and will break easily if you flex it too much or put too much pressure on it.

Remove the screws on each side of the serial and parallel port plugs (J2 and J4). You are now ready to remove the main board. Remove the four screws that retain it and lift it out.

Let's do the simple stuff first. Carefully lift out the CPU. You may use a small flat screwdriver to gently pry it out at both ends. *Important*: Notice which way the notch on one end of the chip is pointing. All chips must face in the same direction.

Install your new Z80B chip in the socket, being careful to point its notch in the right direction. You also have to be careful not to bend any of the pins as you push them in.

Next, change the monitor ROM chip (U47) in the same way. Install the new chip from Micro C.

Cut three 16-inch lengths of your wire now, one of each of three colors. Solder these to the switch first. Do a little planning on this one. The single-pole, double-throw switch has three terminals. The center one is common to both sides. Solder the black wire to the common.

For an easy mnemonic, decide now how you plan to install the switch; that is, determine which way it will point when you want the fast speed, but don't install it yet. I like to toggle the switch up for 5 MHz and down for 2.5 MHz. Solder the red wire to the bottom post and the green wire to the top post if you are using a toggle switch. Reverse this order for a slide switch.

It Gets Hairy

Now it is time for the only part of the job that gets hairy. You have to alter your circuit board. This is a job that requires delicate soldering. If you aren't too sure of your skills in this department, you should consider some alternatives—ask a friend who solders well, practice on some old junk circuit boards (easy to get at flea markets and computer swap meets). Or you could just forget the whole idea.

Solder the black wire to the front side (nearest the edge) of resistor R26. Make the connection on the solder spot where the trace crosses from resistor R26 on its way to pin 4 of IC U86 (Photo 2). Do all these connections on the bottom side of the board, being careful not to overheat anything.

> You can make most software perform twice as fast, but *do not* attempt to change speeds in the middle of an operation—you'll send your operating system to another planet.

Solder your green wire (for the stock 2.5 MHz low speed) to the base of the socket for U86 at pin 4. Solder the red wire (for the 5 MHz high speed) to the base of pin 5 of U86. See the anatomy lesson above if you've forgotten how to count pins.

Now you're ready for the last bit of surgery. Use a small screwdriver or pocket knife to break the trace connecting C81 to pin 4 of the U86. Simply scratch a little metal off the board. This forces the current to flow through your switch.

Once you are done and the solder has cooled, you're ready to put the computer back together. Gather your new wires together and support them so they don't pull loose or get against other parts of the board. Carefully install the main board, reversing the removal procedure. Attach the wires you unplugged. Install your new two-speed switch in a vent slot at the back of the case. I found that the third slot away from the reset switch is a good location.

Now you're ready to put the lid back on and try your new hot rod. Do your first experimentation with disks and data that aren't too important. After you're sure everything is working right, you can go ahead and use the high speed almost all the time.

What Are The Advantages?

There are several advantages to adding this two-speed modification to the Kaypro. Obviously, you can make most software perform twice as much work in the same amount of time. The Kaypro is more likely to keep up with a fast touch-typist (not I, says the little monkey).

This is the only speed upgrade I have seen for the Kaypro that offers the following advantages:

- It doesn't need to introduce any "wait states" into the operating system so the monitor and/or drives can catch up. A wait state is an extra cycle added to the operating system so the CPU won't get ahead of the other components. The new ROM chip makes wait states unnecessary.
- It costs less than $50 to make the upgrade. All the other Kaypro speed-up kits I have seen cost more than $100.
- It offers you the choice of two speeds. While most programs operate fine at 5 MHz, there are some that refuse to cooperate. You have the option of running them at the expected 2.5 MHz speed. You can still run any older software you may have without doctoring it needlessly.
- A nice plus is that this modification doesn't take up any extra space inside your machine. The other kits usually have a small PC board that mounts in place of, or near, the CPU, taking up space you may need for other modifications (like the Co-Power 88 board needed for IBM compatibility).

Warning!

There is something to avoid. When using your Kaypro, *do not* attempt to change speeds in the middle of a program. The glitch introduced by flipping the switch will promptly send your operating system to another planet. You will lose anything that was in memory because the only way to recover is with a system reset.

There is only one thing you may not like. Most of the other speed-up kits on the market are installed by simply plugging them in to replace your CPU. This isn't a kit and you must assemble the pieces yourself. If you're not averse to a little careful assembly, it isn't hard.

Spoil Yourself

Overall, this is a good modification and worth the effort. Once you've been spoiled with the faster system, you will wonder why Kaypro didn't do this itself (it would have been so easy to manufacture the computer with this built in). You'll be reluctant to ever slow the system down again.■

Go on line in the world's fastest growing technology.

NEW! DATA COMMUNICATIONS TRAINING FROM NRI

Practical training includes computer, modem, test instruments, and access to exclusive NRI communications network.

Satellites . . . microwave . . . fiber optics . . . dedicated land lines. Suddenly the world is communicating in a new and different way, via digital data systems. People talking to computers . . . computers to computers . . . information is stored, retrieved, and relayed in nanoseconds.

Industry, opportunities to triple

Data and telecommunications is already a $150 billion industry and is expected to triple over the next five years. One typical company has grown from $85 million to $650 million . . . a 765% growth since 1978 alone. The need for qualified technicians to install, maintain, and service this enormous investment in high-tech equipment is tremendous even now. Opportunities and salaries can go nowhere but up and up.

NRI will train you at home

You can learn at home in your spare time to become a data communications technician with NRI at-home training. NRI will start you with the basics, build upon your knowledge with easy-to-follow, bite size lessons to take you into the world of digital data communications. You'll learn what it takes to work on satellite, microwave, fiber optic, and telephone data links.

And you'll learn at your own comfortable pace, without classroom pressures or evenings away from your family. Over the past 70 years, NRI has taught the latest high-tech skills to almost 2 million students to become the world's largest and most successful school of its kind.

Hands-on training includes computer, modem, breakout box and much more

NRI takes you far beyond "book learning." As part of your course, you receive plenty of practical hands-on training that gives you real-world skills. You get the Radio Shack Color Computer, with 16K memory to teach you the systems and language of data communications plus you get an operating modem to let you tie in with world-wide communications networks.

You build your own RS-232C interface breakout box, an indispensable installation and trouble-shooting instrument you'll use throughout your career. You receive a professional digital multimeter and the NRI Discovery Lab, where you construct solid-state circuits and demonstrate practical applications of the theory you've learned.

Exclusive NRI data network

You'll learn what data communications is all about by actually becoming part of an operating network. You'll go on line to "talk" to your instructor, take your final exam by computer link, communicate with other NRI students and leave messages on the NRI "bulletin board."

As part of your course, you'll also receive membership in THE SOURCE [sm], a regular $100 value. A phone call ties you into computers loaded with instant news, stock quotes, electronic mail, educational programs, games, even discount shopping and travel reservations.

Move into the future, send for Free Catalog

You can't find training like this anywhere else . . . only NRI trains you at home for an exciting and rewarding career in the brilliant new world of Data Communications. Mail the coupon right now for our big catalog of high-tech electronic careers showing all the equipment you get, detailed lesson descriptions, and career opportunities. Look it over and decide where you want your future to grow. Act now. There's a real need for trained data communications technicians.

Training includes all this equipment you keep . . . 16K computer, modem, breakout box, digital multimeter and the exclusive NRI Discovery Lab.

TRS-80 is a trademark of the Radio Shack division of Tandy Corp.
SM a service mark of Source Telecomputing Corp., a subsidiary of the Reader's Digest Association, Inc.

NRI Schools
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue
Washington, D.C. 20016
We'll give you tomorrow.

All Career courses approved under GI bill.
☐ Check for details

CHECK ONE FREE CATALOG ONLY

- ☐ Data Communications
- ☐ Computer Electronics with Microcomputers
- ☐ Color TV, Audio, and Video System Servicing
- ☐ Electronics Design Technology
- ☐ Digital Electronics
- ☐ Communications Electronics • FCC Licenses • Mobile CB • Aircraft • Marine
- ☐ Industrial Electronics
- ☐ Basic Electronics
- ☐ Small Engine Servicing
- ☐ Appliance Servicing
- ☐ Automotive Servicing
- ☐ Auto Air Conditioning
- ☐ Air Conditioning, Heating, Refrigeration, & Solar Technology
- ☐ Building Construction

Name (Please Print) Age

Street

City/State/Zip

Accredited by the Accrediting Commission of the National Home Study Council 172-034

The Key To a Better Atari

Dissatisfied with the Atari 400's membrane keyboard? If so, you can replace it in a jiffy with the Inhome B Key.

By Marc I. Leavey

From the beginning, I had a love-hate relationship with my Atari 400 computer. I loved the available programs, the software, the graphics—in short, just about everything that makes the Atari unique. What I hated was the keyboard. That's why I was very interested when I saw the first advertisements for the Inhome B Key 400, a replacement keyboard for the Atari 400.

Hustling down to my local dealer, I was promptly turned back and sent home. Just as soon as a shipment of B Key keyboards came in, they sold out, sometimes within hours. But I firmly resolved that this was one product I must have and I persevered. A week or so later, I was finally notified that a shipment had arrived.

Bringing Home the New Addition

Arriving back home, I unpacked the small box. Here's what I found: a printed circuit board assembly with a ribbon connector and keyswitches soldered in place, keycaps on a sticky board and an instruction manual. The board is nicely done, with all the Atari keyboard switches represented. The supplied keycaps are all double-injected molded, clearly marked for all Atari functions. The keyboard strongly resembles the one I had used many years ago on my first terminal, a TVT-II, except for the Atari functions on the keys.

The major steps of installation are a snap. I only had to pop the cover off my original Atari 400, unplug the old keyboard, plug the new one in and replace the cover. Place the keycaps on the switches and the modification is done. What you have is an almost full-stroke keyboard as an integral part of the Atari 400.

Address correspondence to Marc Leavey, 6 Jenny Lane, Pikesville, MD 21208.

The Atari 400 keyboard before surgery.

The Atari with the Inhome B Key keyboard in place.

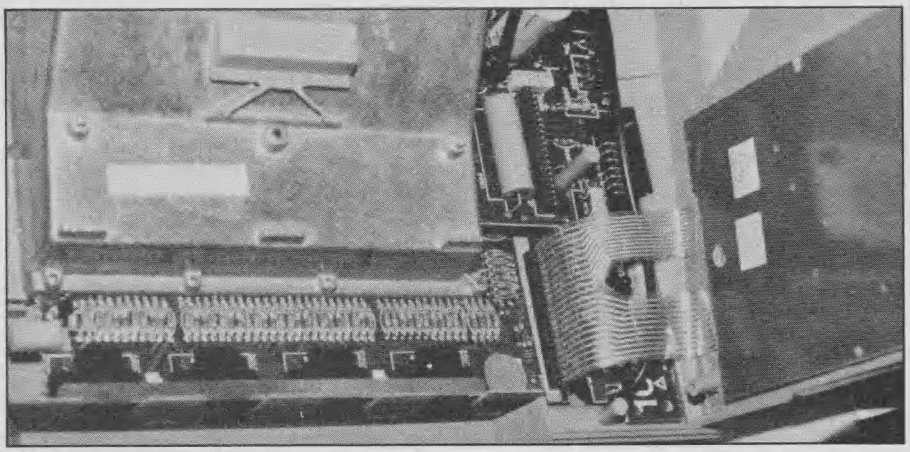
Problems during assembly are mostly limited to inserting the end of the ribbon cable in the keyboard socket. The Atari keyboard cable terminates with small flat prongs that slip into the header on the main Atari board. The B Key cable ends in little round wires, which have to be coaxed a bit in order to enter the socket neatly. I found that by angling the cable, I could slip them in almost one at a time, from back to front. Be sure the cable is seated all the way before buttoning things back up.

Keyboard Feelings

The keyboard has a nice feel to it. The keys do not travel as far as some; that's why I call it an "almost" full-stroke keyboard, but it's a whale of an improvement over the membrane keyboard of the virgin 400. The only comments I can make about the key layout, which is slightly different from the original Atari in order to accommodate the larger keys, are that the return key is the same size as other keys, making it a bit hard to find, and that the Atari logo key is replaced by one that reads Inhome. When I first saw that I thought that this key was for "inserting" text or "homing" the cursor.

Since these functions are not indigenous to the Atari keyboard, I took another look and realized that the Inhome key inverted the text just like the Atari logo key did. Small point, but a source of confusion. Placing the return key down one row, where the asterisk (*) key is now, would have allowed the use of a double-width key for that vital function. However, doing that would have messed up the cluster of four keys, the asterisk being one, which control cursor position in the control mode. I can understand the rationale for not wanting to disturb that group.

The Inhome B Key 400, manufactured in Canada, appears to be the neatest replacement keyboard for the Atari 400 I have seen. It's inexpensive, housed totally within the Atari case and can be installed in about 15 minutes without soldering or drilling. If desired, the old keyboard can always be replaced, returning the Atari 400 to its original condition, although why anyone would want to do that is beyond me. If your local dealer doesn't stock the B Key, more information may be obtained by writing Inhome Software, Inc., 2485 Dunwin Drive, Unit 8, Mississauga, Ontario, Canada L5L 1T1.■



The Atari keyboard cable is the transparent ribbon cable on the right that plugs into the header on the main printed circuit board.

Give Life to Terminal Programs

Tired of the frustration caused by using dumb terminal programs, the author decided to write his own smart terminal program in MBasic for his Kaypro II.

By L.E. Bradley

Terminal programs let microcomputers function as remote terminals for communication with each other or with larger mainframe computers. All terminal programs for microcomputers permit two-way communication, but the availability of additional features varies widely.

Dumb terminal programs can only receive and transmit data. These no-frills programs lack features to store incoming data for later use. Smart terminal programs, on the other hand, offer a variety of options that allow entire programs, text files or other data to be sent or received and retained. Programs or text can be captured in the memory of the receiving computer and then stored on disk for later use.

A dumb terminal program, Term, is part of the bundled software included with the Kaypro II computer. Term can be used to demonstrate limitations of programs that do no more than send and receive characters; once lines scroll off the screen, they are lost forever. The frustration of needing data that had long since scrolled off the display, combined with the high cost of smart terminal programs, prompted me to develop my own solution to the problem.

My first attempt to smarten the machine language Term program proved a formidable task; I soon decided to scrap the project and write my own smart terminal program using MBasic, which is also included with the Kaypro II.

Limitations of Basic

The primary problem with Basic terminal programs is their slow execution speed compared to machine language programs. Limitations are imposed on the number of activities that can be accomplished in loops that get and send characters that are rapidly streaming in. For example, a Basic program can get an incoming character and display it, but it will miss some characters if required to perform many other tasks in the same loop. By the time a Basic program can complete a number of tasks and get back to the input port for the next character, that character may already be lost and replaced by the next one in line.

Furthermore, some activities require more time than others. Even machine-language programs must temporarily halt communication while disk files are accessed.

Another problem with Basic programs is that the interpreter gobbles up nearly 32Kb of your system's memory. This leaves little space for a buffer in which to store incoming characters.

Get with the Program

My Basic terminal program can send and receive data and uses a

```
10 DEFINT A,I,X,C                        'variable names for integers
20 DIM A(15000)                          'dimension array for buffer
30 I = 1                                 'initialize subscript pointer
40 PRINT CHR$(26)                        'Kaypro clear screen command
50   WHILE X$ <> CHR$(27)                'Loop until escape key is pressed
60     X$ = INKEY$                       'X$ will be key pressed
70     IF X$ = "" THEN 100               'if no keypress then check input
80     OUT 4,ASC(X$)                     'send character code out
90     PRINT X$;                         'and print it
100    WHILE INP(6) AND 1                'look at status port for character
110        X=INP(4)                      'get that character
120      IF I > 14985 THEN 250           'is the buffer full?
130      IF X < 30 THEN GOTO 170         'is the code a line feed?
140         PRINT CHR$(X);               'print valid ASCII codes
150         A(I) = X                     'store each one in the array
160         I = I + 1                    'next array element ready
170      IF X <> 10 THEN GOTO 230        'if not a linefeed ignore it
180         PRINT                        'if a linefeed show on screen
190         A(I) = 10                    'save the linefeed in array
200         A(I+1) = 13                  'and add a carriage return
210         I = I + 2                    'add 2 for LF and CR
220       IF I > 14985 THEN GOTO 250     'is buffer full (15000)?
230   WEND                               'end of character in loop
240 WEND                                 'end of character out loop
```

Listing 1.

Address correspondence to L.E. Bradley, Department of Social Behavior, 301 East Hall, University of South Dakota, Vermillion, SD 57069.

15,000-element integer array to store incoming data as it is displayed. This program can keep up with data coming into the Kaypro's serial port at 300 bits per second (bps). The program stores the characters' ASCII codes, one at a time, in the array. By incrementing the array subscript as each new character is detected, it is possible to save up to 15,000 characters of text from a session.

Once the text is loaded into the array, you can print it, display it, or save it in a disk file for later use. Since many slower printers cannot keep up with even a 300 bps terminal program without losing characters at the end of a line, this program provides one means to print a telecommunications session.

Originally, I used a string array to accumulate one line of text, then store it under a single subscript; each element of the array contained a single line of input. That version of the program lost incoming characters because more string manipulation was required.

MBasic's processing speed allows the use of 300 bps communications; eliminating some statements and, consequently, some features, speeds things up a bit.

Any Port in a Storm . . .

The Kaypro and many other computers send and receive data through ports. Two ports are used by my program: a data port and a status port. The data port is where the program goes to get and send data; a status port is used to tell the program when a new character is waiting. Examples 1–4 show the importance of using the status port in conjunction with a data port.

Example 1 demonstrates how Basic can be used to access data coming into a computer through the serial port.

Technical manuals for most computers usually identify the specific port numbers used for serial data transfer. Two ports are used by the Kaypro and many other computers for sending and receiving data. The program must check the status port to determine if it is time to go to the data port for a character.

If the status port is not checked and characters are taken only from the input port, there is no way to know that a new character has been transmitted by the other computer. Example 1 demonstrates how characters may be obtained from the input port without verifying that a character is indeed waiting. The program just grabs whatever is at the port and prints it without determining if it has already read and printed that character.

If the word EXAMPLE has been transmitted to your computer running the program shown in Example 1, the word appears as EEEEEXXXXXAAA AAMMMMMPPPPPLLLLLEEEEE.

The repetition of characters demonstrates the need to let the computer know when each character has been received and is ready to be picked up by the program and printed. To do this, the program needs some form of handshaking, or acknowledgement, which tells the receiving system that a new character has been transmitted and is therefore ready to be printed.

One handshaking technique involves checking another port, called the status port, to determine if another character has been sent to your computer through the serial port. You can establish some form of handshaking between the input ports and the program by using a Wait statement, which halts execution of the program until the value at the port changes, telling you that a character is waiting.

For the Kaypro, when bit 1 of the status port (port 6) is set, your program can go to port 4 to get the next character. If the status bit is not set, the program drops out of the loop without executing line 20. It can now properly receive and display all characters that have been transmitted to the computer. The new example program now looks like Example 2.

Although the program now faithfully produces all characters that are available at your serial port, you are still unable to transmit data *from* your computer. As it stands, the program locks up the Kaypro at the Wait statement when the transmission ends and no new character appears at port 4 to reset the status port. Consequently, you cannot transmit characters out of your locked-up computer to respond to the message that you have received.

```
10 X=INP(4)
20 PRINT CHR$(X);
30 GOTO 10
```

Example 1.

```
10 WAIT 6,1                    ;wait until bit one of
                                port 6 is set
20 PRINT CHR$(INP(4));         ;print the character
30 GOTO 10                     ;jump to line 10 and
                                start again
```

Example 2.

```
10 WHILE INP(6) AND 1< > 0
20   PRINT CHR$(INP(4));
30 WEND
40 GOTO 10
```

Example 3.

```
80  OUT 4,ASC(X$)+128
110 X=INP(4) AND 127
```

Example 4.

```
250  PRINT CHR$(26)
260  X$=""
270  PRINT "1) RETURN TO TERMINAL PROGRAM"
280  PRINT "2) VIEW BUFFER ON SCREEN"
290  PRINT "3) PRINT BUFFER "
300  INPUT "YOUR CHOICE  ";R
310  ON R GOTO 40,400,340
320  GOTO 250
330  PRINT "Insert paper and turn printer on "
340  INPUT "Press return when ready          ";R$
350  LPRINT
360  FOR K = 1 TO I
370     LPRINT CHR$(A(K));
380  NEXT K
390  GOTO 250
400  REM *** PRINT BUFFER TO SCREEN ***
410  FOR K = 1 TO I
420     PRINT CHR$(A(K));
430  NEXT K
440  INPUT"PRESS RETURN TO CONTINUE ";R$
450  GOTO 250
```

Listing 2.

What You Need

You need a way to test the status port without locking up your computer when no characters are waiting. You need some other Basic statement to test the single bit of the status port to determine when a character is waiting there. Using an And statement, you can test a single bit of any variable or port. According to information provided in the Kaypro user's manual, a single bit of the byte at the status port will be set (a binary 1) if a character has been transmitted to the computer. Since it is bit 1 of the status port that you want to test, simply And the value at the status port with the value 1.

If the result on the test is true (bit 1 is set) then the program can print and store the incoming character in its buffer. A While construct with the And statement returns a true condition each time the first bit of the status port is set. The program does not hang up, as is the case when it used the Wait statement. The operations in the loop are performed only when the status bit of port 6 is set. If it isn't set, the program can bail out of the loop and do other things.

Your program (Example 3) now faithfully prints all characters coming into the computer. You can make it transmit data from the computer by using another series of Basic routines to send characters to the output port. The routine that picks up and transmits characters from the keyboard can be placed in another loop, into which you nest the character-receive loop just described. If you provide for keyboard input, you can type special characters that cause the program to jump from its transmit and receive loops into a part of the program that can print or save the stored characters.

> The program now faithfully prints *all* characters coming from the computer.

The simplest way to input from the keyboard without halting program execution is to use the INKEY$ function to pick up a keystroke while the program is looping and scanning the status port for incoming characters. Handshaking for the output characters has none of the problems encountered while receiving data because the program needs only to send each character and be done with it.

The next refinement is one to filter unprintable control codes. These codes can shut down your printer, clear the screen, reboot your system or do other undesirable things. A simple if-then test suffices; your complete program appears in Listing 1.

The program in Listing 1 provides the means to send and receive ASCII characters through the serial port of the Kaypro II computer, providing that the remote computer uses eight data bits and no parity. If the sending (host) computer uses even parity and seven data bits, change lines 80 and 110 of the program to those shown in Example 5.

Finally, what the program needs is a routine that prints the buffer. To add a buffer-print routine simply add a for-next loop to print the ASCII characters for the codes stored in array A(n). This part of the program is shown in Listing 2.

To use the terminal program, connect the modem to the serial port at the rear of the Kaypro and establish contact with the host system. After the remote computer is accessed and the carrier-detect light is lit on the modem, log onto the host system. If your display doesn't show the characters that you type, be sure that you have typed line 90 correctly. If you see two characters for every one that you type, delete line 90. (Be sure to save your program if you modify it.)

If the host system doesn't recognize your input, check the parity of the other system and that produced by your program. For even parity, add 128 to the value you send to port 4 (line 80); you should also strip the parity bit from the input by Anding the input byte with 127 (line 110). Above all, make sure that the bit rate of the host system is 300 bps (the Kaypro default).

Once the program is typed, saved and perhaps modified somewhat, you can modify it to perform other tasks. A few enhancements that you can add to the program include an expanded menu with provisions for listing the disk directory, saving files on disk and transmitting files to the host computer.■

NEW FROM Wayne Green Books

You Can Design Electronic Circuits The Easy Way

With Programs for Electronic Circuit Design

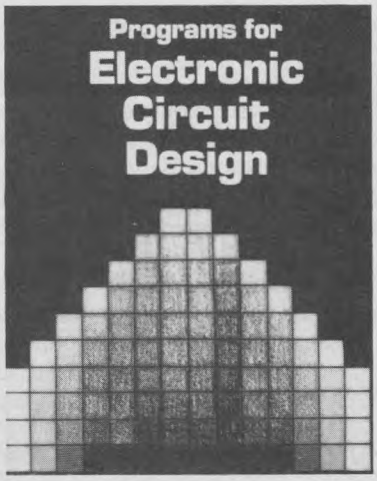
Use your microcomputer and *Programs for Electronic Circuit Design* help you select the correct value for each component in an electronic circuit. The programs are adaptable to most microcomputer systems and are also available on disk for the Apple, IBM and TRS-80. Each chapter contains a program involving several different types of applications that use the same electronic components.

The thirteen programs can be used individually, or they can be combined, using a master menu, as explained in the book. The book provides documentation for the programs, and some of the programs refer to schematics and diagrams in the book.

If you have an elementary knowledge of electronics and familiarity with the basic components, you will find *Programs for Electronic Circuit Design* an invaluable aid in circuit design.

Topics covered include:

- resistor, capacitor and inductor circuits
- circuits that have combinations of resistors, capacitors and inductors



- series and parallel circuits
- Ohm's law computations
- voltage dividers
- impedance and frequency computations
- phase angles
- operational amplifiers
- 555 timers
- filters
- transistor circuits

Programs for Electronic Circuit Design
David Leithauser
$14.95 ISBN 0-88006-068-9
softcover 7 by 9 approx. 100 pp. 1984
BK7400
Book and disk packages. $24.95
Apple CC740011
IBM PC CC740012
TRS-80 CC740013

To order, call toll-free 1-800-258-5473 for credit card orders. Or mail your order with check or money order or complete credit card information to: Wayne Green Books, Retail Sales, Peterborough, NH 03458. Please include $1.50 for the first item and $1.00 for each additional item for shipping and handling.

Send me _____ copies of *Programs for Electronic Circuit Design* (BK7400) at $14.95 each.
Book and disk packages: _____ Apple (CC740011) _____ IBM PC (CC740012)
_____ TRS-80 (CC740013) at $24.95 each.
Shipping and handling is $1.50 for the first item; $1.00 for each additional item.

Card #_____ MC bank #_____

☐payment enclosed ☐MasterCard ☐Visa ☐Am. Ex.

Expiration date_____ Signature_____

Name_____

Address_____

City_____ State_____ Zip_____

Wayne Green Books, Peterborough, NH 03458

343B7P

Turning the Tables

A table search is a useful programming technique, but there are many searching methods from which to choose. The author describes several searching techniques to help you find the best for your purposes.

By John Pantone

The table search is one of the most common, and most useful, techniques used in programming. Compilers and assemblers must look up symbolic addresses, mnemonic codes and many other names in tables.

Database systems have to find keys, search for names and extract data based on relations between one key and another. Operating systems must verify commands, look up filenames in directories and so on.

```
10 '
20 '                        LINEAR SEARCH EXAMPLE
30 '
40 DIM TABLE$(15)
50 DATA apl,cobol,pl/i,fortran,c,pascal,lisp,modula,ada
60 DATA jovial,logo,smalltalk,assembler,forth,pilot
70 '
80 '              Fill TABLE$ with valid entries
90 '
100 FOR I=1 TO 15
110 READ TABLE$(I)
120 NEXT I
130 CLS
140 '
150 '             Begin lookups (ask for target)
160 '
170 PRINT
180 I=1
190 INPUT "Target: ";T$
200 IF T$="quit" THEN END
210 '
220 '             Loop here until target is found in table
230 '             or all 15 entries have been tested.
240 '
250 IF T$=TABLE$(I) THEN 320
260 I=I+1
270 IF I>15 THEN 370:' All entries have been tested, target not found.
280 GOTO 250:' Continue until target found, or all entries tested.
290 '
300 '             Target was found
310 '
320 PRINT T$," found. Entry number: ";I
330 GOTO 170:' Go back and ask for another target.
340 '
350 '             Target was not found after all 15 entries were tested
360 '
370 PRINT T$," not found after 15 tries."
380 GOTO 170:' Go back and ask for another target.
```

Listing 1. An example of a linear search.

While it is often sufficient to verify the presence of a word in a table, there are many other reasons for using table searches. For example, you could use a table search to find a person's name in a table, then use the position in the table (element number) to retrieve a phone number from another table; this is called an indexed table. If you found the name Smith in element 14 of your table of names, Mr. Smith's phone number could be found in element 14 of the phone number table. You could, of course, continue this process with other pieces of information, such as Smith's age, employment history and so on.

An extension of this technique can be applied to files. Instead of reading a huge personnel file until you find Smith, you could search for Smith in your table of names and find that Mr. Smith's records are in record 14 of the personnel file. This would let you read only one record (using direct access) of the large personnel file.

Tackling the Terminology

Before examining some of the methods for table searches, you should be familiar with the terminology. When looking for a specific word in a table,

Address correspondence to John Pantone, Harwood Drive, Danbury, CT 06810.

that word is called a target. If you're able to locate the target in the table, you have a hit. If not, you have a failure, or a miss.

Because of the value of table searches and the ingenuity of programmers, many types of table searches have been developed. Let's look at the advantages and disadvantages of four of these techniques.

The Linear Search

This is the first technique that most programmers think of when attempting to write a table search. The method consists of reading each entry in the table until either the target is found or the end of the table is reached.

The advantages of this technique are mostly related to ease of programming and expandability. It is simple to implement such a table search and new entries can be added easily; simply tack them onto the end of the table. An example of a linear search is shown in Listing 1.

The major disadvantage is obvious; it could take a long time to find a target, especially in a large table. Even worse, if the target is not in the table, the entire table must be searched before you can be sure that you have failed (a miss).

However, if the table is small, if the system doesn't have to be fast (as is the case in batch-oriented systems), or if the table must be frequently updated, this technique can be handy.

Ordered Search

A closely related table search method is called the ordered search. This method is really a linear search, except that the table is arranged so targets assumed to be frequently searched for are at the top of the table, and those assumed to be rarely searched for are at the bottom.

The advantages and disadvantages of the straight linear search also apply to the ordered search, with the addition of one advantage and one disadvantage.

The advantage is that if you guess correctly, most of the targets searched for in the table will be found more quickly than in a linear search, since they will be at the top of the table. The disadvantage is that since the table has to be kept in an ordered sequence, adding new items can become difficult.

An ordered search example is shown in Listing 2. (This table favors languages that are traditionally compiled over those usually interpreted. It generally takes longer to search in a Pascal program than it does in a Fortran program.)

```
10 '
20 '                    ORDERED SEARCH EXAMPLE
30 '
40 DIM TABLE$(15)
50 '
60 '             Table is ordered to favor lookup of languages
70 '             which are traditionally compiled directly to
80 '             native machine code.
90 '
100 DATA ada,assembler,c,cobol,fortran,jovial,pl/i
110 DATA apl,forth,pascal,lisp,logo,modula,pilot,smalltalk
120 '
130 '              Fill TABLE$ with valid entries
140 '
150 FOR I=1 TO 15
160 READ TABLE$(I)
170 NEXT I
180 CLS
190 '
200 '             Begin lookups (ask for target)
210 '
220 PRINT
230 I=1
240 INPUT "Target: ";T$
250 IF T$="quit" THEN END
260 '
270 '             Loop here until target is found in table
280 '             or all 15 entries have been tested.
290 '
300 IF T$=TABLE$(I) THEN 370
310 I=I+1
320 IF I>15 THEN 420:' All entries have been tested, target not found.
330 GOTO 300:' Continue until target found, or all entries tested.
340 '
350 '             Target was found
360 '
370 PRINT T$," found. Entry number: ";I
380 GOTO 220:' Go back and ask for another target.
390 '
400 '             Target was not found after all 15 entries were tested
410 '
420 PRINT T$," not found after 15 tries."
430 GOTO 220:' Go back and ask for another target.
```

Listing 2. An example of an ordered search.

```
LINEAR SEARCH

Target: ? cobol
cobol              found. Entry number:  2

Target: ? lisp
lisp               found. Entry number:  7

Target: ? forth
forth              found. Entry number:  14

Target: ? frog
frog               not found after 15 tries.

Target: ? quit
```

Example 1. The results of a linear search.

```
ORDERED SEARCH

Target: ? cobol
cobol              found. Entry number:  4

Target: ? lisp
lisp               found. Entry number:  11

Target: ? forth
forth              found. Entry number:  9

Target: ? frog
frog               not found after 15 tries

Target: ? quit
```

Example 2. The results of an ordered search.

Hashing

If you use a formula of some kind to change your target from a word to a number and then use that number to represent the position in your table, you have a simple hash-based search. The word hash refers to the use of a formula that changes a word or phrase into a number.

This table-searching technique is fast, but difficult to program. The biggest advantage of this method is its speed. Since you can look directly at the correct element of the table, regardless of its size, you can also quickly determine if the target isn't in the table. An example of a hash-based table search is shown in Listing 3.

In this program, I used a simple formula to arrive at the hash number: Each letter of the target is changed to a number from one to 26 and then added together to yield the hash number for the word. In this case "ada" hashes to six and "c" to three. By manually calculating the hash values of each entry, the largest number yielded would be 101; that is the size I chose for the table.

```
10 '
20 '                        TABLE LOOKUP BY HASHING
30 '
40 DEFINT A-Z
50 DIM TABLE$(101):' 101 found to be sufficient by manual calculations
60 '
70 '                      set each element of table to "<empty>"
80 '
90 FOR I=0 TO 101:TABLE$(I)="<empty>":NEXT I
100 '
110 '             this string is used to calculate the hash number
120 '
130 CHARS$="abcdefghijklmnopqrstuvwxyz"
140 '
150 '             These are the valid entries
160 '
170 DATA apl,cobol,pl/i,fortran,c,pascal,lisp,modula,ada
180 DATA jovial,logo,smalltalk,assembler,forth,pilot
190 '
200 '                       fill table
210 '
220 FOR I=1 TO 15
230 '             calculate hash number for this entry
240 K=0
250 READ ENTRY$
260 FOR J=1 TO LEN(ENTRY$)
270 K=K+INSTR(CHARS$,MID$(ENTRY$,J,1))
280 NEXT J
290 '             place the entry in the element calculated
300 TABLE$(K)=ENTRY$
310 NEXT I
320 '
330 '                    Start doing lookups
340 '
350 CLS
360 INPUT "Target: ";TARGET$
370 IF TARGET$="quit" THEN END
380 '
390 '    Hash target to determine table entry. Notice that this is done
400 '    the same way as was done when loading the table.
410 '
420 K=0
430 FOR I=1 TO LEN(TARGET$)
440 K=K+INSTR(CHARS$,MID$(TARGET$,I,1))
450 NEXT I
460 '
470 IF K>101 THEN K=0:' don't allow k to be > than the length of table$
480 '
490 '       Actual lookup just requires that we inspect the entry
500 '       at the hashed value. Note that if the target equals the entry
510 '       we have a 'hit', if the table entry is 'empty' then we have
520 '       a 'miss', if the table entry is not 'empty' but the target and
530 '       entry aren't equal there has been a 'collision'.
540 '
550 IF TABLE$(K)=TARGET$ THEN PRINT TARGET$,"found. Element ";K
560 IF TABLE$(K)<>TARGET$ THEN PRINT TARGET$,"not found."
570 IF TABLE$(K)<>TARGET$ AND TABLE$(K)<>"<empty>" THEN PRINT "COLLISION wi
    th table entry: ";TABLE$(K);" element: ";K
580 PRINT:GOTO 360:' Go back and get a new target for lookup.
```

Listing 3. An example of the hashing techique for table searches.

Listing 4. An example of the binary chop.

```
10 '
20 '                        TABLE LOOKUP BY BINARY CHOP
30 '
40 CLS
50 DEFINT A-Z
60 DIM TABLE$(17)
70 '
80 '             Set up table. Note that *bot* (beginning of table)
90 '             and *eot* (end of table) are present to force
100 '            all valid entries to be 'between' them.
110 '            Note that this causes ada to be element number 2, not 1.
120 '
130 DATA *bot*,ada,apl,assembler,basic,c,cobol,forth,fortran
140 DATA lisp,logo,modula,pascal,pilot,pl/i,smalltalk,*eot*
150 '
160 '            Load TABLE$ with valid entries, and count the
170 '            number of entries, for use later.
180 '
190 TLENGTH=1
200 READ TABLE$(TLENGTH)
210 IF TABLE$(TLENGTH)="*eot*" THEN 250 ELSE TLENGTH=TLENGTH+1:GOTO 200
220 '
230 '            Begin lookup
240 '
250 PRINT
260 INPUT "Target: ";TARGET$
270 IF TARGET$="quit" THEN END
280 TRIES=1
290 '                      set up first guess and top/bot
300 GUESS = TLENGTH/2
310 TOP=1:BOT=TLENGTH
320 '                      loop while guessing
```

More →

Now that some of the mystery in table searches has been removed, you can make an informed decision when choosing a search method.

The example shown has one favorable attribute: none of the 15 entries in the table hash to the same number. Although this makes the example easier to follow, it is an unusual circumstance. If you added "aaa" to the table, it would hash to the number three, which is already used by "c." This same-number hash is called a collision, and is one of the major drawbacks to this method of table searching.

There are many ways to overcome this problem, most of which are just as complicated to program as the table search itself. One way to avoid this collision problem is to choose a different formula, one which will uniquely hash all of the entries.

Another way to avoid the problem is to use a "rehash." Rehashing involves hashing a word over again (usually by a slightly different formula) if a collision occurs while placing it in the table. In this case, the number of rehashes necessary to finally insert the element into the table must be saved, since, when searching for the word later, you must allow for at least that number of rehashes and reexaminations of the table before you can be assured that the target isn't in the table.

You may have already noticed the other major drawback of the hashing method; the elements of the table are spread out over the table, with a lot of empty space between elements. The example has only 15 valid entries, yet needs 101 table elements of storage. The other examples needed only 15 elements in the table.

Fine tuning the formula used for hashing to produce the fewest number of collisions and retries, while at the same time producing the smallest table possible, is an art. Unless the major advantage of this method (speed) is necessary, it's seldom used.

Binary Chop

The last type of table search to be examined is the "binary-chop" or

"bchop" method. Perhaps the best example of this technique is to refer to a game called Hi-Lo. This game, you may recall, asks you to guess the number that the computer has chosen. You're told if your guess is too high or too low. Normally the range of numbers is from one to 100.

The fastest way to guess the number is to guess half way between your last guess and the top or bottom of the range. Each time you guess past the number, alter the top or bottom of the range and continue. For example, your first guess would always be 50 (for a range of one to 100).

The computer replies TOO HIGH. You assign your new bottom to 50 and guess half way between the top and bottom—25.

The computer then replies TOO HIGH. Again assign a new bottom to 25 and guess half way between the top and bottom—12. (Ignore halves; this is an integer game.)

The computer replies TOO LOW. You now assign your new top to 12 and guess half way between the top and bottom—18. This process continues until the correct number is targeted.

A little inspection reveals that this guessing method never takes more than seven guesses to hit the right number. (The game usually restricts you to six guesses just to be frustrating!)

The only thing you have to do to adapt this type of search to words instead of numbers is to keep your table in sorted order. Too high becomes too high in the sort order and too low is too low in the sort order.

This type of search is an effective compromise between a linear search and the hashing method. It is considerably easier to program than a full-blown hashing search, yet it is quite a bit faster than a linear search. In fact, doubling the size of the table only requires one more guess.

The major disadvantage of this table search is the necessity of keeping the table in sorted order. It's time consuming to add a new element, since the entire table then has to be resorted. Listing 4 shows an example of the binary-chop search.

Listing continued.

```
330 TRIES=TRIES+1
340 IF ABS(TOP-BOT) = 1 THEN 480;' Table 'used up', target not found.
350 IF TABLE$(GUESS)=TARGET$ THEN 450;' current guess is correct, target fo
    und.
360 IF TABLE$(GUESS)>TARGET$ THEN 410;' current guess 'too high'
370 '                    guess too low
380 TOP=GUESS
390 GUESS=(TOP+BOT)/2
400 GOTO 320;' Keep looking
410 '                    guess too high
420 BOT=GUESS
430 GUESS=(TOP+BOT)/2
440 GOTO 320;' Keep looking
450 '                    got it
460 PRINT TARGET$," found after ";TRIES-1;" tries.  Table element ";GUESS
470 GOTO 250;' Go back and try another target.
480 '                    not found
490 PRINT TARGET$," not found after ";TRIES-1;" tries."
500 GOTO 250;' Go back and try another target.
```

Table Search Comparison

	Simple linear search	Ordered linear search	Hashed search	Binary chop search
Ease of programming	1	1	3	2
Table initialization	1	2	3	2
Memory utilization	1	1	3	1
Speed (hit)	3	3	1	2
Speed (miss)	3	3	1	1
Ease of adding elements	1	3	2	3

NOTE: 1—good, 2—fair, 3—poor

Table 1. A summary of the major features of various table searches.

```
BINARY CHOP (BCHOP)

Target: ? cobol
cobol            found after  3  tries.  Table element  7

Target: ? lisp
lisp             found after  4  tries.  Table element  10

Target: ? forth
forth            found after  4  tries.  Table element  8

Target: ? fros
fros             not found after  5  tries.

Target: ? quit
```

Example 4. The results of a binary chop.

Things Are Looking Up

There are many ways to search for words in a table. Each method has advantages and disadvantages. The major features of each of these table searches are summarized in Table 1.

Now that some of the mystery in table searches has been removed, you can make an informed decision when choosing a search method.■

```
HASHED LOOKUP

Target: ? cobol
cobol            found. Element  47

Target: ? lisp
lisp             found. Element  56

Target: ? forth
forth            found. Element  67

Target: ? fros
fros             not found.

Target: ? c
c                found. Element  3

Target: ? aaa
aaa              not found.
COLLISION with table entry: c element:

Target: ? quit
```

Example 3. The results of the hashing sorting routine.

A Cheap Way To Spool Around

For less than $70, you can build an expandable spooler for your parallel port printer!

By Dennis C. Fait

If you're low on cash, but would love to have a spooler for your parallel port printer, read on and I'll tell you how you can build your own expandable spooler for less than $70.

By calling it *expandable* I mean that you can start with only 2Kb of RAM and slowly add to the spooler's memory as your budget allows.

By the term spooler I mean a small dedicated microcomputer that allows your computer to dump data to your printer without the computer—or you—having to sit around and wait.

Normally, when your computer is printing a listing or document it feeds the data to the printer, usually a line at a time; then it waits until the printer has printed the data before sending more. This amounts to a waste of time and wear and tear on your disk drives when they turn on and off as your computer accesses them for more data to print.

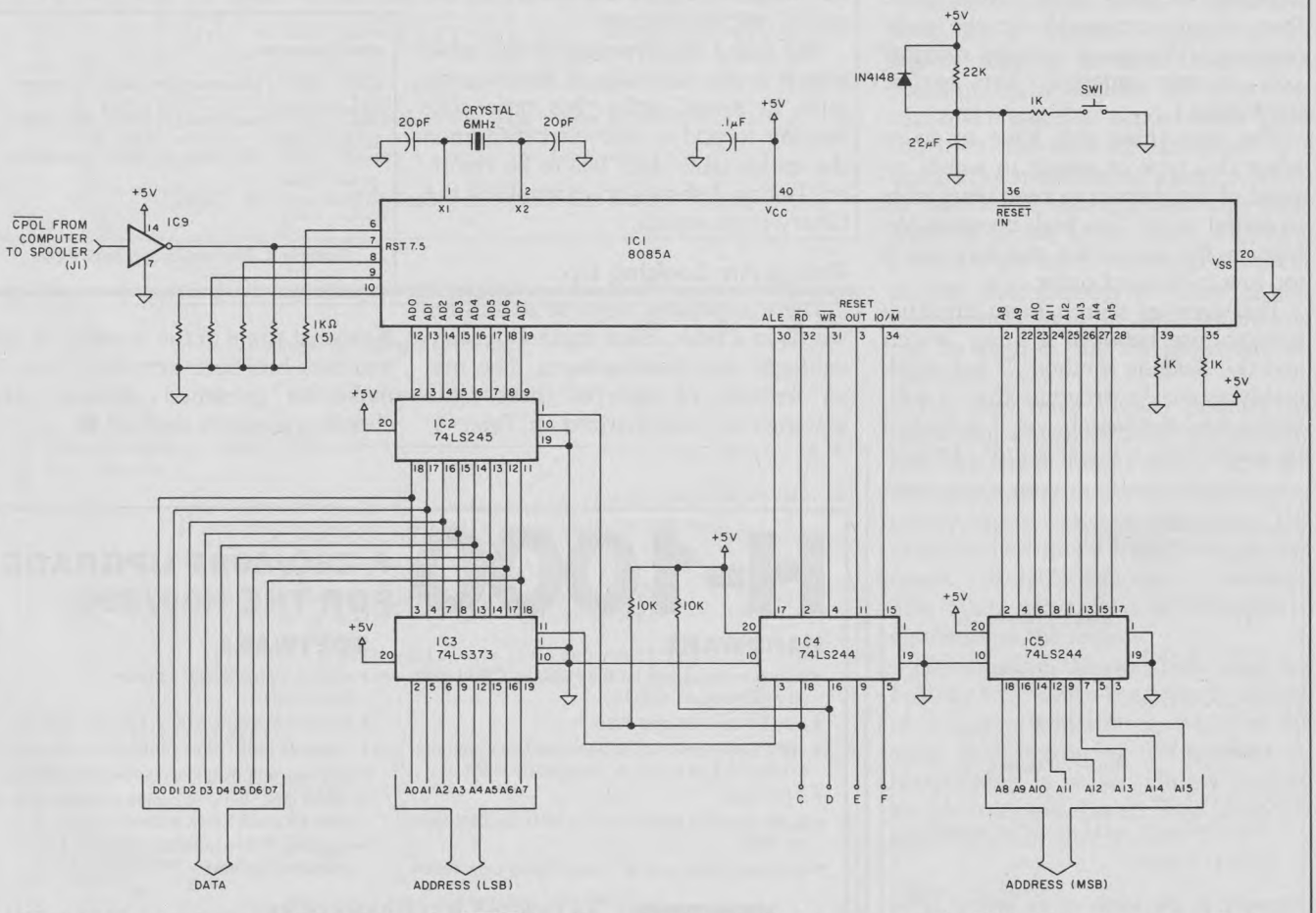


Fig. 1. 8085A microprocessor with buffers. Note that $\overline{CPOL}$ from the computer is inverted by IC9 and sent to the CPU's RST7.5 interrupt on Pin 7.

A spooler, then, will take the data from your computer, store it in its own memory buffer and feed it out to your printer as the printer requires it. The spooler does the waiting...not your computer.

You can build this spooler for an initial cost of about $70, not including memory. With all memory installed (60Kb of buffer RAM) the final cost can exceed $200, but before you reach that point, one of us will figure out how to use denser and cheaper memory chips (refer to my suggestions at the end of this article).

Choosing the Chips

The memory chips used in this project are 2016 static RAMs (of the equivalent 6116 CMOS version). They're easy to work with (address decoding is a snap) and they're pin-for-pin compatible with 2716 EPROMs.

I selected the 8085A as the microprocessor because it's easy to use, requires only a positive five-volt supply and has vectored interrupts. (And besides that, I have a disk-based 8080/8085 assembler that makes software development an easy task.)

Other hardware pieces required for this project (see Table 1) include various buffering ICs, one 8255 I/O device, several address decoding chips, and, of course, the memory chips and a power supply.

I built the processor circuit and the memory circuit on two separate perfboards, mounted one above the other horizontally and put them and the power supply into a cabinet.

I then ran the I/O lines from the processor board to two DB-25 connectors on the cabinet. (Your computer/printer combination may require different connectors.)

A Look at the Circuit

All address, data and control signals from the 8085A are buffered by IC2,4 and 5 (Fig. 1). While these buffering chips might not be necessary in every case, they do prevent some whacky things from happening to the signals. IC3 (a 74LS373) separates the data signals from the eight least-significant address signals. These 16 signals are multiplexed within the 8085A and use the same eight pins. It's necessary, then, to demultiplex them.

Although I've used a 6 MHz crystal for the clock, almost any frequency from about 3 MHz to 6.144 MHz may be used. With the 6 MHz crystal installed, using 450 ns memory devices, I have yet to experience any difficulties in reading or writing to memory.

SW1 allows you to manually reset the spooler. The remaining circuitry around SW1 automatically resets it when you turn on the power.

Two 2114 static memory chips (Fig. 2), which total 1Kb of RAM and are addressed at FC00H, are used for the system stack and scratchpad memory. These may be eliminated if you build a complete 64Kb spooler. IC8 decodes the address lines for these two chips.

IC10 (a 74LS154, Fig. 3) and IC11 (if you go for the full 64Kb) decode the addresses for the 2016 memory chips. Pin 1 of IC10 is the chip-select signal for IC14, a 2716 EPROM that contains the spooler firmware, and consequently is addressed at 0000H.

Pins 2–11 and 13–17 are the chip-select signals for the next 15 RAM chips. Pins 1–11 and 13–17 of IC11 select the next group of RAM. Pin 17, however, is not used if the 2114 stack memory chips are used.

IC12 (an 8255) and its decoding chip, IC13, provide three I/O ports (and one control port) for getting the data from your computer and sending it to the printer (Fig. 4). Port A, pins 1–4 and 37–40, is configured as an input port for data from your computer. Port B, pins 18–25, is an output port to the printer.

Port C, pins 10–13 and 14–17, is split into two 4-bit nibbles; one is for inputting handshaking signals from your computer and printer, and the other is for outputting control signals to those two devices. Configuring the 8255 ports for these functions is handled by software.

$\overline{\text{PXDR}}$, the printer ready signal, is fed to one bit of Port C; $\overline{\text{SPOL}}$ and $\overline{\text{SXDR}}$ from Port C are generated by software and sent to the printer and computer, respectively.

$\overline{\text{SPOL}}$ tells the printer that a byte of data is ready and $\overline{\text{SXDR}}$ tells the computer that the spooler is busy. $\overline{\text{CPOL}}$, the signal from the computer that it has a data byte ready, is inverted by IC9 and fed to interrupt RST7.5 at pin 7 of the 8085A.

One other signal is fed to Port C. It

J1
(Input Port From Computer)

Pin	Signal	Description
1	Chassis ground	Common to spooler power supply
2	Signal ground	
16	$\overline{\text{SXDR}}$	Spooler ready for data from computer
17	$\overline{\text{CPOL}}$	Computer data ready
18	DI7	Data input lines
19	DI6	Data input lines
20	DI5	Data input lines
21	DI4	Data input lines
22	DI3	Data input lines
23	DI2	Data input lines
24	DI1	Data input lines
25	DI0	Data input lines

J2
(Output Port To Printer)

Pin	Signal	Description
1	Chassis ground	Common to spooler power supply
2	Signal ground	
16	$\overline{\text{PXDR}}$	Printer ready for data from spooler
17	$\overline{\text{SPOL}}$	Spooler data ready for printer
18	DO7	Data output lines
19	DO6	Data output lines
20	DO5	Data output lines
21	DO4	Data output lines
22	DO3	Data output lines
23	DO2	Data output lines
24	DO1	Data output lines
25	DO0	Data output lines

Table 1. DB-25 pin designations. This is the DB-25 connector pin-out for the Centronics parallel port signals used by the spooler. Your computer/printer combination may use different connectors.

Address correspondence to Dennis C. Fait, PO Box 22, Slippery Rock, PA 16057.

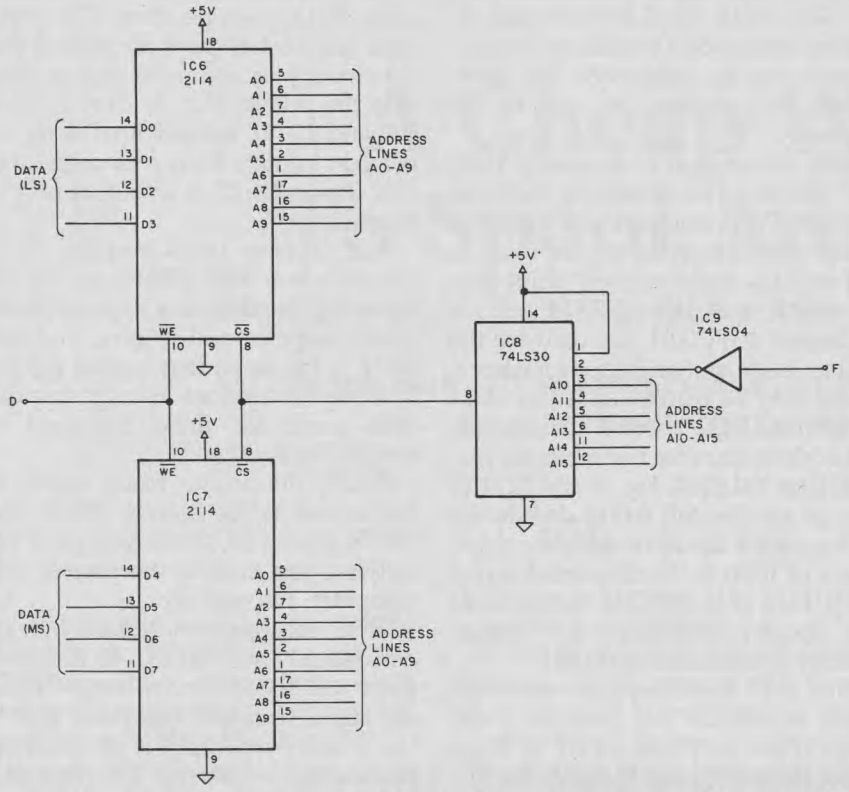


Fig. 2. Stack and scratchpad memory. The two 2114 static memory devices are addressed at FC00H by IC8. This 1Kb of memory is used by the 8085A for the stackpointer and as temporary storage for any program variables.

> The DATIN routine must be located at 3CH, the RST 7.5 vector, otherwise things will go haywire.

comes from SW2, which pulls pin 15 high; when released, it allows the pin to be drawn low.

I've used this signal to instruct the software to send printer-dependent code to my MX-80. Two unused lines on Port C, pins 14 and 16, can also be used in this fashion if you wish. All you have to do is add the switches and resistors, and have the program check these two bits during initialization.

Make Room for Software Goodies

In Listing 1 is the program to run all this hardware. It's less than 300 bytes long, leaving a lot of room in your EPROM for other goodies.

Basically, the program accepts data

SOFTWARE DESCRIPTIONS

TPM (TPM I) - $80 A Z80 only operating system which is capable of running CP/M programs. Includes many features not found in CP/M such as independent disk directory partitioning for up to 255 user partitions, space, time and version commands, date and time, create FCB, chain program, direct disk I/O, abbreviated commands and more! Available for North Star (either single or double density), TRS—80 Model I (offset 4200H) or II, Versafloppy I, or Tarbell I.

TPM-II - $125 An expanded version of TPM which is fully CP/M 2.2 compatible but still retains the extra features our customers have come to depend on. This version is super FAST. Extended density capability allows over 600K per side on an 8" disk. Available preconfigured for Versafloppy II (8" or 5"), Epson QX-10, Osborne II or TRS-80 Model II.

CONFIGURATOR I

This package provides all the necessary programs for customizing TPM for a floppy controller which we do not support. We suggest ordering this on single density (8SD).
Includes: TPM-II ($125), Sample PIOS (BIOS) SOURCE ($FREE), MACRO II ($100), LINKER ($80), DEBUG I ($80), QED ($150), ZEDIT ($50), TOP I ($80), BASIC I ($50) and BASIC II ($100)
$815 Value **NOW $250**

CONFIGURATOR II

Includes: TPM-II ($125), Sample PIOS (BIOS) SOURCE ($FREE), MACRO II ($100), MACRO III ($150), LINKER ($80), DEBUG I ($80), DEBUG II ($100), QSAL ($200), QED ($150), ZTEL ($80), TOP II ($100), BUSINESS BASIC ($200) and MODEM SOURCE ($40) and DISASSEMBLER ($80)
$1485 Value **NOW $400**

MODEL I PROGRAMMER

This package is only for the TRS-80 Model I. Note: These are the ONLY CDL programs available for the Model I. It includes: TPM I ($80), BUSINESS BASIC ($200), MACRO I ($80), DEBUG I ($80), ZDDT ($40), ZTEL ($80), TOP I ($80) and MODEM ($40)
$680 Value **NOW $175**

MODEL II PROGRAMMER

This package is only for the TRS-80 Model II. It includes: TPM-II ($125), BUSINESS BASIC ($200), MACRO II ($100), MACRO III ($150), LINKER ($80), DEBUG I ($80), DEBUG II ($100), QED ($150), ZTEL ($80), TOP II ($100), ZDDT ($40), ZAPPLE SOURCE ($80), MODEM ($40), MODEM SOURCE ($40) and DISASSEMBLER ($80)
$1445 Value **NOW $375**

BASIC I - $50, a 12K - basic interpreter with 7 digit precision.

BASIC II - $100. A 12 digit precision version of Basic I.

BUSINESS BASIC - $200. A full disk extended basic with random or sequential disk file handling and 12 digit precision (even for TRIG functions). Also includes PRIVACY command to protect source code, fixed and variable record lengths, simultaneous access to multiple disk files, global editing, and more!

ACCOUNTING PACKAGE - $300. Written in Business Basic. Includes General Ledger, Accounts Receivable/Payable and Payroll. Set up for Hazeltine 1500 terminal. Minor modifications needed for other terminals. Provided in unprotected source form.

MACRO I - $80. A Z80/8080 assembler which uses CDL/TDL mnemonics. Handles MACROs and generates relocateable code. Includes 14 conditionals, 16 listing controls, 54 pseudo-ops, 11 arithmetic/logical ops, local and global symbols, linkable module generation, and more!

MACRO II - $100. An improved version of Macro I with expanded linking capabilities and more listing options. Also internal code has been greatly improved for faster, more reliable operation.

MACRO III - $150. An enchanced version of Macro II. Internal buffers have been increased to achieve a significant improvement in speed of assembly. Additional features include line numbers, cross reference, compressed PRN files, form feeds, page parity, additional pseudo-ops, internal setting of time and date, and expanded assembly-time data entry.

DEVELOPER I

Includes: MACRO I ($80), DEBUG I ($80), ZEDIT ($50), TOP I ($80), BASIC I ($50) and BASIC II ($100)
$440 Value **NOW $150**

DEVELOPER II

Includes: MACRO II ($100), MACRO III ($150), LINKER ($80), DEBUG I ($80), DEBUG II ($100), BUSINESS BASIC ($200), QED ($150), TOP II ($100), ZDDT ($40), ZAPPLE SOURCE ($80), MODEM SOURCE ($40), ZTEL ($80), and DISASSEMBLER ($80).
$1280 Value **NOW $350**

DEVELOPER III

Includes: QSAL ($200), QED ($150), BUSINESS BASIC ($200), ZTEL ($80) and TOP II ($100)
$730 Value **NOW $300**

COMBO

Includes: DEVELOPER II ($1280), ACCOUNTING PACKAGE ($300), QSAL ($200) and 6502X ($150)
$1930 Value **NOW $500**

LINKER - $80. A linking loader for handling the linkable modules created by the above assemblers.

DEBUG I - $80. A tool for debugging Z80 or 8080 code. Disassembles to CDL/TDL mnemonics compatible with above assemblers. Traces code even through ROM. Commands include Calculate, Display, Examine, Fill, Goto, List, Mode, Open File, Put, Set Wait, Trace, and Search.

DEBUG II - $100. A superset of Debug I. Adds Instruction Interpreter, Radix change, Set Trap/Conditional display, Trace options, and Zap FCB.

6502X - $150. A 6502 cross assembler. Runs on the Z80 but assembles 6502 instructions into 6502 object code! Similar features as our Macro assemblers.

QSAL - $200. A SUPER FAST Z80 assembler. Up to 10 times faster than conventional assemblers. Directly generates code into memory in one pass but also to offset for execution in its own memory space. Pascal like structures: repeat...until, if...then...else, while...do, begin...end, case...of. Multiple statements per line, special register handling expressions, long symbol names, auto and modular assembly, and more! This one uses ZILOG Mnemonics.

QED - $150. A screen editor which is both FAST and easy to learn. Commands include block delete, copy, and move to a named file or within text, repeat previous command, change, locate, find at start of line, and numerous cursor and window movement functions. Works with any CRT having clear screen, addressable cursor, clear to end of line, clear to end of screen, and 80X24.

ZTEL - $80. An extensive text editing language and editor modelled after DEC's TECO.

ZEDIT - $50. A mini text editor. Character/line oriented. Works well with hardcopy terminals and is easy to use. Includes macro command capability.

TOP I - $80. A Text Output Processor for formatting manuals, documents, etc. Interprets commands which are entered into the text by an editor. Commands include justify, page number, heading, subheading, centering, and more.

TOP II - $100. A superset of TOP I. Adds: embedded control characters in the file, page at a time printing, selected portion printing, include/merge files, form feed/CRLF option for paging, instant start up, and final page ejection.

ZDDT - $40. This is the disk version of our famous Zapple monitor. It will also load hex and relocatable files.

ZAPPLE SOURCE - $80. This is the source to the SMB ROM version of our famous Zapple monitor. It can be used to create your own custom version or as an example of the features of our assemblers. Must be assembled using one of our assemblers.

MODEM - A communication program for file transfer between systems or using a system as a terminal. Based on the user group version but modified to work with our SMB board or TRS-80 Models I or II. You must specify which version you want.

MODEM SOURCE - $40. For making your own custom version. Requires one of our Macro Assemblers.

DISASSEMBLER - $80. Does bulk disassembly of object files creating source files which can be assembled by one of our assemblers.

HARDWARE

S-100 — SMB II Bare Board $50. "System Monitor Board" for S-100 systems. 2 serial ports, 2 parallel ports, cassette interface, 4K memory (ROM, 2708 EPROM, 2114 RAM), and power on jump. When used with Zapple ROM below, it makes putting a S-100 system together a snap.
Zapple ROM $35. Properly initializes SMB I/II hardware, provides a powerful debug monitor.

IBM PC — Big Blue Z80 board $595. Add Z80 capability to your IBM Personal Computer. Runs CP/M programs but does not require CP/M or TPM. Complete with Z80 CPU, 64K add on memory, serial port, parallel port, time and date clock with battery backup, hard disk interface, and software to attach to PC DOS and transfer programs. Mfr'd by QCS.
50% Discount on all CDL software ordered at the same time as a Big Blue (and for the Big Blue).

APPLE II — Chairman Z80 $345. Add Z80 capability to your Apple II/II Plus computer. Runs CP/M programs with our more powerful TPM. Includes 64K memory add on (unlike the competition this is also useable by the 6502/DOS as well as the Z80), TPM, QSAL assembler, QED Screen Editor, and Business Basic. Mfr'd by AMT Research.
Apple Special $175. Buy the Apple Z80 Developer at the same time as the "Chairman" and pay only $175 instead of $325.

APPLE Z80 DEVELOPER

Includes: 6502X ($150), MACRO II ($100), MACRO III ($150), QSAL ($200), QED ($150), LINKER ($80), DEBUG I ($80), DEBUG II ($100), ZDDT ($40) and BUSINESS BASIC ($200)
VALUE: $1250 **NOW $325**
$175 when purchased with AMT "Chairman" Board

DISK FORMATS

When ordering software specify which disk format you would like.

CODE	DESCRIPTION
8SD	8" IBM 3740 Single Density (128 bytes/26 sectors/77 tracks)
8DD	8" Double Density (256 bytes/26 sectors/77 tracks)
8XD	8" CDL Extended Density (1024 bytes/8 sector/77 traceks 616K)
5SD	5.25" Single Density (TRS80 Model I, Versafloppy I, Tarbell I)
5EP	5.25" Epson Double Density
5PC	5.25" IBM PC Double Density
5XE	5.25" Xerox 820 Single Density
5OS	5.25" Osborne Single Density
5ZA	5.25" Z80 Apple (Softcard compatible)

TPM INFO When ordering TPM I or II, in addition to Disk Format, please specify one of the following codes:

	CODE	DESCRIPTION
TPM I:	NSSD/H	North Star Single Density for Horizon I/O
	NSSD/Z	North Star Single Density for Zapple I/O
	NSDD/H	North Star Double Density for Horizon I/O
	NSDD/Z	North Star Double Density for Zapple I/O
	TRS80-I	TRS-80 Model I (4200H Offset)
	TRS80II	TRS-80 Model II
	VI8	Versafloppy I 8"
	VI5	Versafloppy I 5.25"
TPM-II:	VII8	Versafloppy II 8" (XD)
	VII5	Versafloppy II 5.25"
	TRS80II	TRS-80 Model II (XD)

ORDERING INFORMATION:

VISA/MasterCard/C.O.D.
Call or Write With Ordering Information....

OEMS:

Many CDL products are available for licensing to OEM's. Write to Carl Galletti with your requirements.

Dealer Inquiries Invited.

For Phone Orders ONLY Call Toll Free...

1-(800) 458-3491 (Except Pa.)

Ask For Extension #15

For information and Tech Queries call
(609) 599-2146

Prices and Specifications subject to change without notice.
TPM, Z80, CP/M, TRS80 are trademarks of CDL, Zilog, DRI and Tandy respectively.

Computer Design Labs

342 Columbus Avenue/Trenton, NJ 08629

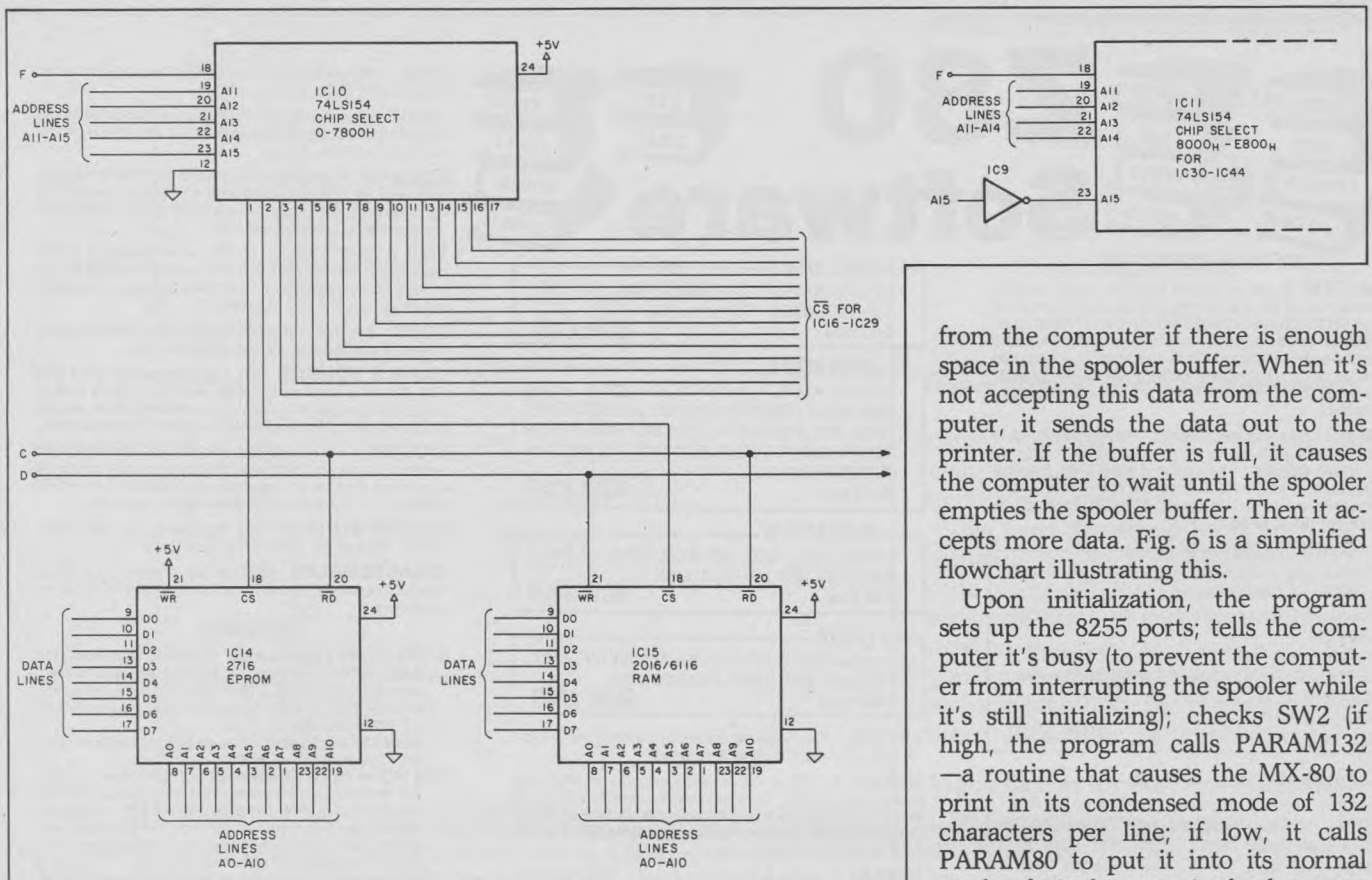


Fig. 3. EPROM and 2016 memory. IC14, a 2716 EPROM, contains the spooler's firmware and is addressed at 0000H. The data buffer begins at 0800H with IC15, and extends through EFFFH with IC44. Chip select signals for the EPROM and 2016s are derived from IC10 and 11, 74LS154s.

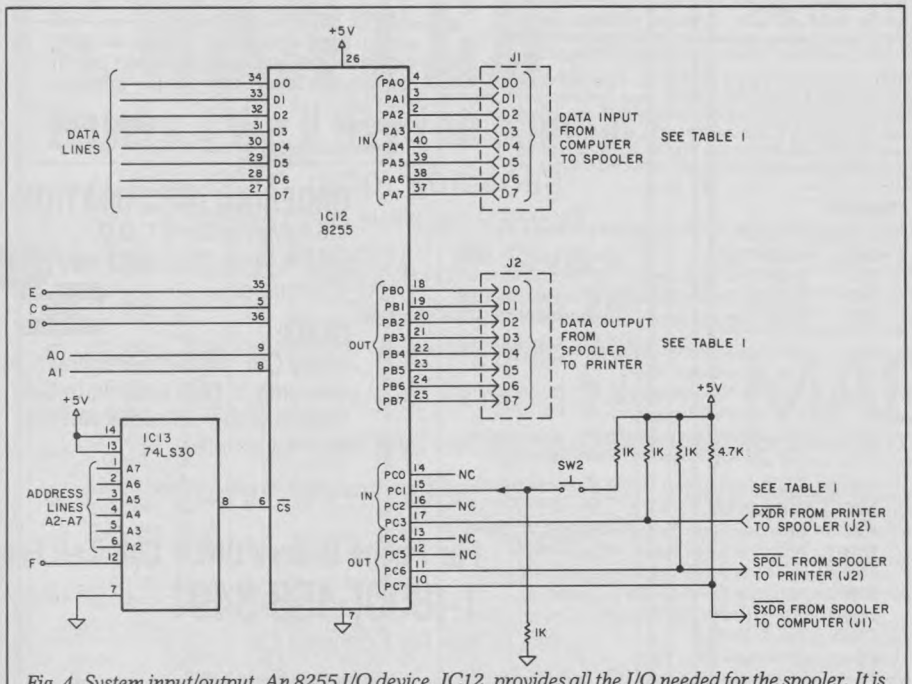


Fig. 4. System input/output. An 8255 I/O device, IC12, provides all the I/O needed for the spooler. It is addressed at FCH through FFH by IC13. Port A, FCH, is an input port for data from the computer; Port B, FDH, is an output port for data going to the printer; and Port C, FEH, is used both for generating and receiving control signals from the computer and printer. Three bits of Port C can be used for sensing switches that allow you to send printer-dependent code to the printer. The port addressed at FFH is the 8255 control port. It is used to configure the 8255 and to toggle Port C's output bits on and off.

from the computer if there is enough space in the spooler buffer. When it's not accepting this data from the computer, it sends the data out to the printer. If the buffer is full, it causes the computer to wait until the spooler empties the spooler buffer. Then it accepts more data. Fig. 6 is a simplified flowchart illustrating this.

Upon initialization, the program sets up the 8255 ports; tells the computer it's busy (to prevent the computer from interrupting the spooler while it's still initializing); checks SW2 (if high, the program calls PARAM132—a routine that causes the MX-80 to print in its condensed mode of 132 characters per line; if low, it calls PARAM80 to put it into its normal mode of 80 characters); checks to see how much RAM is available for its buffer (a feature that allows you to add RAM to the spooler as you can afford it); and sets two pointers to the beginning of the buffer.

One pointer, the B and C registers, points to the next address to store the next byte coming from the computer, while the other pointer, the H and L registers, points to the next byte to send to the printer.

Registers D and E hold the address-plus-one of the highest address in the buffer. This is used to check to see if the buffer is full. Obviously, if the value of BC equals the value of DE, the buffer is full. If BC equals HL, all the data in the buffer has been sent to the printer.

Whenever the computer has data for the spooler and $\overline{\text{SXDR}}$ is not active (the spooler is not busy), $\overline{\text{CPOL}}$ becomes active, causing an interrupt. Because we've attached $\overline{\text{CPOL}}$ to RST7.5, the 8085A will store a return address on the stack and will call the routine at 3CH, which services the interrupt. The DATIN routine must be located at 3CH, the RST7.5 vector, otherwise things will go haywire.

DATIN checks if there is room in the buffer and, if there is, accepts the data from the computer and puts it into the buffer. If there is no room in the buffer, it calls MEMFULL, causing

the computer to wait until the spooler empties the buffer to the printer.

If the data you're dumping to the spooler/printer is less than the amount of buffer memory in the spooler, MEMFULL will never be called. After storing the byte in the buffer, DATIN returns control to the part of the program that was interrupted.

That routine was probably the loop that begins at TIMER and ends at either PRINT1 or PRINT2. This loop continuously checks to see if there is data in the buffer to send to the printer; if there is, it prints that data.

The timer routine apparently is needed to give the interrupts a head-start in filling the buffer before the spooler starts to send the buffer contents to the printer. I say "apparently" because until I added TIMER the loop would not work.

Construction

Any wiring technique will work because layout and wire lengths are unimportant. However, you may find that wire wrapping is the quickest. Be sure to liberally use .1 μf bypass capacitors on the power bus throughout the circuit.

Table 1 shows the usual Centronics parallel port pin-out for DB-25 connectors.

Cutting Chip Count and Cost

One practical change to make to the spooler's hardware is to use 4164 dynamic RAM chips. This would reduce the chip (and socket!) count and the overall cost of the spooler. The first 2Kb of the 4164s, however, would have to be phantomed to make room for the 2716 EPROM. One way of doing this is to use the EPROM chip select signal to Tri-state the data output lines of the 4164s.

Also, if 4164s are used, the 2114s in Fig. 2 wouldn't be needed and the memory-checking routine in the program would have to be deleted. (Consequently, the D and E registers would have to be initiated with the highest buffer address you intend to use. Allowing room for the stack at FFFFH, D and E could be set at something around FF00H.)

You may want to add some other printer-dependent routines to the two I'm using. (Like most of my projects, I'll probably keep on adding things to this one.)

Another nifty idea would be to add another set of I/O ports for a second printer. That way, with the throw of one single-pole double-throw switch, you could have the software select one of two printers to send its buffer contents to. (If it weren't necessary to laboriously file out another hole for a DB-25 connector, I'd have done this already.)

Like I said, that 2716 has lots of room for more programming.■

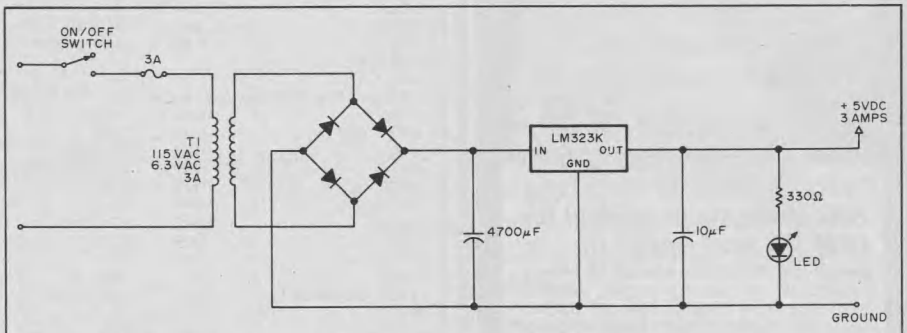


Fig. 5. Typical power supply. The LM323K is a 5V 3A device and should be mounted on an external heatsink.

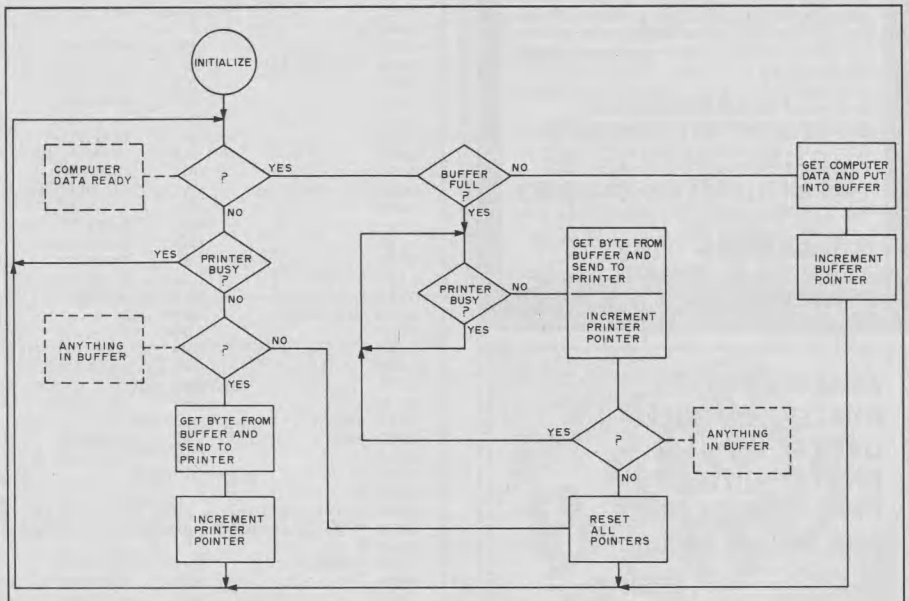


Fig. 6. Spooler flowchart. It contains two minor loops: 1) print while receiving data from computer; and, 2) print entire buffer.

Listing 1. Control program for spooler. It resides in a 2716 EPROM addressed at 0000H.

```
0010 * FILENAME--> SPOOLINT.S
0020 *
0030 * FIRMWARE FOR PARALLEL PORT PRINTER SPOOLER
0040 *
0050 *     --- USES RST7.5 INTERRUPT ---
0060 *
0070 * By
0080 *   Dennis C. Fait
0090 *   PO Box 22
0100 *   Slippery Rock, Pa. 16057
0110 *
0120 * May 22, 1983
0130 *
0140 *
0150 COMDATA    EQU   0FCH      ;INPUT PORT, COMPUTER DATA
0160 PRINDATA   EQU   0FDH      ;OUTPUT PORT, DATA TO PRINTER
0170 STATUS     EQU   0FEH      ;4-BIT STATUS PORT
0180 CONTROL    EQU   0FFH      ;4-BIT CONTROL PORT
0190 SBUSY      EQU   0FH       ;CONTROL BIT TO COMPUTER, SPOOLER BUSY
0200 SFREE      EQU   0EH       ;CONTROL BIT TO COMPUTER, SPOOLER FREE
0210 SDATA      EQU   0CH       ;CONTROL BIT TO PRINTER, DATA READY
0220 NODATA     EQU   0DH       ;CONTROL BIT TO PRINTER, NO DATA READY
0230 PRIN       EQU   8         ;MASK FOR PRINTER BUSY STROBE
0240 BUTT       EQU   2         ;MASK FOR BUTTON #1
0250 *
0260 *
```

More

Listing continued.

```
0270 *
0280 *
0290 * EPROM containing this program is addressed at 0000H.
0300                ORG     0
0310                MVI     A,00011011B     ;PREPARE RST7.5 MASK
0320                SIM
0330                JMP     INITIALIZE
0340                FILL    0FFH,0FFH
0350 *
0360 *
0370 * The following interrupt routine MUST be addressed at 3CH.
0380                ORG     3CH             ;RST7.5 ROUTINE
0390 DATIN          PUSH.   PSW             ;SAVE THIS
0400                MVI     A,SBUSY         ;TELL COMPUTER SPOOLER IS BUSY
0410                OUT     CONTROL
0420                MOV     A,D             ;BUFFER FULL?
0430                CMP     B
0440                JNZ     GETDATA         ;NO
0450                MOV     A,E
0460                CMP     C
0470                CZ      MEMFULL         ;YES, SO PRINT REST OF BUFFER
0480 GETDATA        IN      COMDATA         ;GET DATA FROM COMPUTER
0490                STAX    B               ;STORE IT IN SPOOLER BUFFER
0500                INX     B
0510                MVI     A,SFREE
0520                OUT     CONTROL
0530                POP     PSW
0540                EI
0550                RET
0560 *
0570 HERE           EQU     $
0580                FILL    0FFH,0FFH       ;FILL WITH SOME FFs
0590 *
0600                ORG     HERE+0FH        ;MAIN PROGRAM
0610 INITIALIZE     DI
0620                MVI     A,91H           ;SET UP 8255 PORTS
0630                OUT     CONTROL
0640                MVI     A,SBUSY         ;TELL COMPUTER THAT SPOOLER IS BUSY
0650                OUT     CONTROL
0660                MVI     A,NODATA        ;TELL PRINTER THAT SPOOLER HAS NO DATA
0670                OUT     CONTROL
0680                LXI     SP,0FFFFH       ;SET STACK POINTER
0690                IN      STATUS
0700                ANI     BUTT            ;BUTTON PRESSED?
0710                CNZ     PARAM132        ;YES
0720                CZ      PARAM80         ;NO
0730 INIT2          LXI     D,0800H
0740                MVI     A,77H           ;LET'S FIND TOP OF BUFFER RAM
0750 MEMCHECK       STAX    D
0760                LDAX    D
0770                CPI     77H
0780                JNZ     INIT3           ;DE=TOP OF RAM+1
0790                INX     D
0800                JMP     MEMCHECK
0810 INIT3          EI
0820 INIT4          LXI     H,0800H         ;HL=NEXT BYTE TO PRINTER
0830                LXI     B,0800H         ;BC=NEXT ADDRESS TO STORE COMPUTER DATA
0840 *
0850 *
0860 * The following loop is to give computer time to download some
0870 * data into spooler's buffer before printer takes off
0880 *
0890 TIMER          MVI     A,SFREE         ;TELL COMPUTER, SPOOLER IS READY NOW
0900                OUT     CONTROL
0910                MVI     A,0FFH
0920 TIME1          DCR     A
0930                JNZ     TIME1
0940 *
0950 * Let's try to print from buffer
0960 PRINT          IN      STATUS
0970                ANI     PRIN
0980                JNZ     PRINT           ;PRINTER IS BUSY
0990                MOV     A,B
1000                CMP     H
1010                JNZ     PRINT1          ;THERE'S SOMETHING IN BUFFER TO PRINT
1020                MOV     A,C
1030                CMP     L
1040                JZ      PRINT2          ;NOTHING IN BUFFER TO PRINT
1050 PRINT1         MOV     A,M             ;GET DATA FROM BUFFER
1060                INX     H
1070                OUT     PRINDATA        ;PRINT IT
1080                MVI     A,SDATA         ;TELL PRINTER DATA IS READY
1090                OUT     CONTROL
1100                INR     A               ;(SAME AS NODATA)
1110                OUT     CONTROL
1120                JMP     PRINT
1130 PRINT2         LXI     H,0800H         ;RESET POINTERS
1140                LXI     B,0800H
1150                JMP     TIMER
1160 *
1170 *
1180 MEMFULL        EQU     $               ;MEMORY FULL, EMPTY BUFFER TO PRINTER
1190                IN      STATUS
1200                ANI     PRIN
1210                JNZ     MEMFULL         ;PRINTER IS BUSY, SO LOOP
1220                MOV     A,M
1230                OUT     PRINDATA
1240                MVI     A,SDATA
1250                OUT     CONTROL
1260                INR     A
1270                OUT     CONTROL
1280                MOV     A,B
1290                CMP     H
```

More →

Listing continued.

```
1300                    JNZ     MEM1            ;MORE TO GO
1310                    MOV     A,C
1320                    CMP     L
1330                    JZ      MEM2            ;BUFFER IS EMPTY
1340 MEM1               INX     H
1350                    JMP     MEMFULL
1360 MEM2               LXI     H,0800H         ;RESET POINTERS
1370                    LXI     B,0800H
1380                    RET
1390 *
1400 * Following routine will send code to MX-80 to
1410 * configure it for 132 characters per line
1420 *
1430 PARAM132           EQU     $
1440                    PUSH    PSW             ;SAVE FLAGS
1450                    CALL    PREADY          ;WAIT TIL PRINTER IS READY
1460                    MVI     A,0FH
1470                    OUT     PRINDATA
1480                    MVI     A,SDATA
1490                    OUT     CONTROL
1500                    INR     A
1510                    OUT     CONTROL
1520                    CALL    PREADY
1530                    MVI     A,7
1540                    OUT     PRINDATA        ;SOUND BUZZER
1550                    MVI     A,SDATA
1560                    OUT     CONTROL
1570                    INR     A
1580                    OUT     CONTROL
1590                    POP     PSW
1600                    RET
1610 * The following routine configures the MX-80 for 80 characters per line
1620 PARAM80            EQU     $
1630                    CALL    PREADY
1640                    MVI     A,12H
1650                    OUT     PRINDATA
1660                    MVI     A,SDATA
1670                    OUT     CONTROL
1680                    INR     A
1690                    OUT     CONTROL
1700                    RET
1710 *
1720 PREADY             IN      STATUS
1730                    ANI     PRIN
1740                    JNZ     PREADY
1750                    RET
1760 *
1770 * END
```

Quantity	Item	Description
1	8085A	Microprocessor
1	74LS245	Buffer
1	74LS373	Demultiplexer
2	74LS244	Buffer
1	6 MHz crystal	System clock
2	20 pF capacitor	
1	1N4148 diode	
2	2114 RAM	Stack memory
2	74LS30	I/O, memory decoding
1	74LS04	Inverter
1	8255	I/O ports
2	74LS154	Memory decoding
1	2716 EPROM	System firmware
30	2016/6116	Buffer RAM
2	N.O. push button switch	Reset, . . .
2	40-pin sockets	
34	24-pin sockets	
4	20-pin sockets	
2	18-pin sockets	
3	14-pin sockets	

Power Supply:

1	6.3VAC 3A transformer	
1	Bridge rectifier	
1	LM323K	three amp regulator
1	SPDT switch	

Misc.:

Resistors, capacitors, LED, zip cord, jumper sockets, connectors, perf board, cabinet and cables.

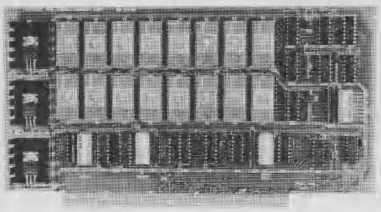
Table 2. Parts list.

DIGITAL RESEARCH COMPUTERS

(214) 271-3538

32K S-100 EPROM CARD

PRICE CUT!



$59.95

USES 2716's

Blank PC Board - $34

ASSEMBLED & TESTED ADD $30

SPECIAL: 2716 EPROM's (450 NS) Are $4.95 Ea. With Above Kit.

KIT FEATURES:

1. Uses +5V only 2716 (2Kx8) EPROM's
2. Allows up to 32K of software on line!
3. IEEE S-100 Compatible.
4. Addressable as two independent 16K blocks.
5. Cromemco extended or Northstar bank select.
6. On board wait state circuitry if needed.
7. Any or all EPROM locations can be disabled.
8. Double sided PC board: solder-masked, silk-screened.
9. Gold plated contact fingers.
10. Unselected EPROM's automatically powered down for low power.
11. Fully buffered and bypassed.
12. Easy and quick to assemble.

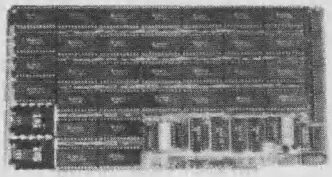
64K S100 STATIC RAM

$199.00 KIT

NEW!

LOW POWER!

RAM OR EPROM!



BLANK PC BOARD WITH DOCUMENTATION $55

SUPPORT ICs + CAPS $17.50

FULL SOCKET SET $14.50

FULLY SUPPORTS THE NEW IEEE 696 S100 STANDARD (AS PROPOSED)

FOR 56K KIT $185

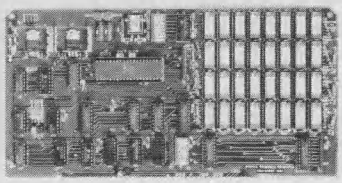
ASSEMBLED AND TESTED ADD $50

FEATURES:

- Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- Fully supports IEEE 696 24 BIT Extended Addressing.
- 64K draws only approximately 500 MA.
- 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- SUPPORTS PHANTOM (BOTH LOWER 32K AND ENTIRE BOARD).
- 2716 EPROMs may be installed in any of top 48K.
- Any of the top 8K (E000 H AND ABOVE) may be disabled to provide windows to eliminate any possible conflicts with your system monitor, disk controller, etc.
- Perfect for small systems since BOTH RAM and EPROM may co-exist on the same board.
- BOARD may be partially populated as 56K.

256K S-100 SOLID STATE DISK SIMULATOR!

WE CALL THIS BOARD THE "LIGHT-SPEED-100" BECAUSE IT OFFERS AN ASTOUNDING INCREASE IN YOUR COMPUTER'S PERFORMANCE WHEN COMPARED TO A MECHANICAL FLOPPY DISK DRIVE.



ADD $50 FOR A & T KIT.

BLANK PCB (WITH CP/M* 2.2 PATCHES ON DISK) $69.95

FEATURES:

- 256K on board, using + 5V 64K DRAMS.
- Uses new Intel 8203-1 LSI Memory Controller.
- Requires only 4 Dip Switch Selectable I/O Ports.
- Runs on 8080 or Z80 S100 machines.
- Up to 8 LS-100 boards can be run together for 2 Meg. of On Line Solid State Disk Storage.
- Provisions for Battery back-up.
- Software to mate the LS-100 to your CP/M* 2.2 DOS is supplied.
- The LS-100 provides an increase in speed of up to 7 to 10 times on Disk Intensive Software.
- Compare our price! You could pay up to 3 times as much for similar boards.

$399.00

#LS-100 (FULL 256K KIT)

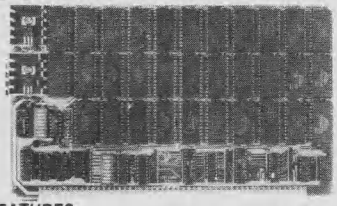
64K SS-50 STATIC RAM

$179.00 (48K KIT)

NEW!

LOW POWER!

RAM OR EPROM!



BLANK PC BOARD WITH DOCUMENTATION $52

SUPPORT ICs + CAPS $18.00

FULL SOCKET SET $15.00

56K Kit $219

64K Kit $249

ASSEMBLED AND TESTED ADD $50

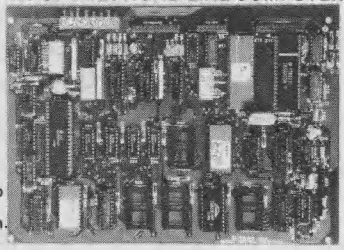
FEATURES:

- Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- Fully supports Extended Addressing.
- 64K draws only approximately 500 MA.
- 200 NS RAMs are standard. (TOSHIBA makes TMM 2016s as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- Board is configured as 3-16K blocks and 8-2K blocks (within any 64K block) for maximum flexibility.
- 2716 EPROMs may be installed anywhere on Board.
- Top 16K may be disabled in 2K blocks to avoid any I/O conflicts.
- One Board supports both RAM and EPROM.
- RAM supports 2MHZ operation at no extra charge!
- Board may be partially populated in 16K increments.

THE NEW ZRT-80

CRT TERMINAL BOARD!

A LOW COST Z-80 BASED SINGLE BOARD THAT ONLY NEEDS AN ASCII KEYBOARD, POWER SUPPLY, AND VIDEO MONITOR TO MAKE A COMPLETE CRT TERMINAL. USE AS A COMPUTER CONSOLE, OR WITH A MODEM FOR USE WITH ANY OF THE PHONE-LINE COMPUTER SERVICES.



FEATURES:

- Uses a Z80A and 6845 CRT Controller for powerful video capabilities.
- RS232 at 16 BAUD Rates from 75 to 19,200.
- 24 x 80 standard format (60 Hz).
- Optional formats from 24 x 80 (50 Hz) to 64 lines x 96 characters (60 Hz).
- Higher density formats require up to 3 additional 2K x 8 6116 RAMS.
- Uses N.S. INS 8250 BAUD Rate Gen. and USART combo IC.
- 3 Terminal Emulation Modes which are Dip Switch selectable. These include the LSI-ADM3A, the Heath H-19, and the Beehive.
- Composite or Split Video.
- Any polarity of video or sync.
- Inverse Video Capability.
- Small Size: 6.5 x 9 inches.

BLANK PCB WITH 2716 CHAR. ROM, 2732 MON. ROM $59.95

SOURCE DISKETTE - ADD $10

SET OF 2 CRYSTALS - ADD $7.50

ADD $50 FOR A & T KIT.

ZRT-80

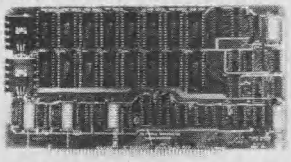
WITH 8 IN. SOURCE DISK! **$129.95** (COMPLETE KIT, 2K VIDEO RAM)

32K S100 EPROM/STATIC RAM

NEW! FOUR FUNCTION BOARD! **NEW!**

EPROM II FULL EPROM KIT $80.00

A&T EPROM ADD $35.00



BLANK PC BOARD WITH DATA $39.95

SUPPORT IC'S PLUS CAPS $23.00

FULL SOCKET SET $18

We took our very popular 32K S100 EPROM Card and added additional logic to create a more versatile EPROM/RAM Board.

FEATURES:

- This one board can be used in any one of four ways:
  - A. As a 32K 2716 EPROM Board
  - B. As a 32K 2732 EPROM Board (Using Every Other Socket)
  - C. As a mixed 32K 2716 EPROM/2K x 8 RAM Board
  - D. As a 32K Static RAM Board
- Uses New 2K x 8 (TMM2016 or HM6116) RAM's
- Fully Supports IEEE 696 Buss Standard (As Proposed)
- Supports 24 Bit Extended Adressing
- 200 NS (FAST!) RAM'S are standard on the RAM Kit
- Supports both Cromemco and North Star Bank Select
- Supports Phantom
- On Board wait State Generator
- Every 2K Block may be disabled
- Addressed as two separate 16K Blocks on any 64K Boundary
- Perfect for MP/M* Systems
- RAM Kit is very low power (300 MA typical)

32K STATIC RAM KIT — $129.95

For RAM Kit A&T - Add $40

TERMS: Add $3.00 postage. We pay balance. Orders under $15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tex Res. add 5% Tax. Foreign orders (except Canada) add 20% P & H. Orders over $50, add 85¢ for insurance.

Digital Research Computers

P.O. BOX 461565 • GARLAND, TEXAS 75046 • (214) 271-3538

ALL SALES ARE MADE SUBJECT TO THE TERMS OF OUR 90 DAY LIMITED WARRANTY. A COPY OF THIS WARRANTY IS AVAILABLE FREE, ON REQUEST.

*TM OF DIGITAL RESEARCH INC. (CALIF.)

WE ARE NOT ASSOCIATED WITH DIGITAL RESEARCH INC. (CALIF.) THE SUPPLIERS OF CPM SOFTWARE

FEATURES THE POWERFUL, THIRD GENERATION, MOTOROLA 6809 PROCESSOR!

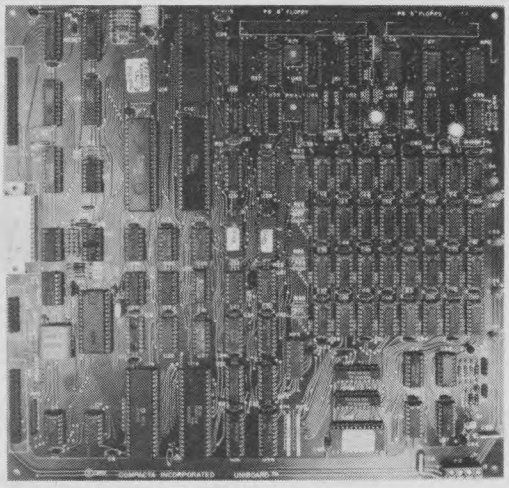
THE 6809 "UNIBOARD"™

SINGLE BOARD COMPUTER KIT

New!

PERFECT FOR COLLEGES, OEM'S, INDUSTRIAL AND SCIENTIFIC USES!

64K RAM! DOUBLE DENSITY FLOPPY DISK CONTROLLER!



BLANK PC BOARD
$99.95
WITH PAL'S, AND TWO EPROMS.

FOR 5-1/4 OR 8 INCH SOURCE DISKETTE ADD $10.

$399.00
COMPLETE KIT! FULLY SOCKETED.

ALL OPTIONS ARE STANDARD. NO EXTRAS TO BUY!

THE COMPACTA UNIBOARD™: Through special arrangement with COMPACTA INC., we are proud to have been selected the exclusive U.S. Mfg. of their new 6809 UNIBOARD™ COMPUTER KIT. Many software professionals feel that the 6809 features probably the most powerful instruction set available today on ANY 8 bit micro. Now, at last, all of that immense computing power is available at a truly unbelievably low price.

FEATURES:

- ★ 64K RAM using 4116 RAMS.
- ★ 6809E Motorola CPU.
- ★ Double Density Floppy Disk Controller for either 5-1/4 or 8 inch drives. Uses WD1793.
- ★ On board 80 x 24 video for a low cost console. Uses 2716 Char. Gen. Programmable Formats. Uses 6845 CRT Controller.
- ★ ASCII keyboard parallel input interface. (6522)
- ★ Serial I/O (6551) for RS232C or 20 MA loop.
- ★ Centronics compatible parallel printer interface. (6522)
- ★ Buss expansion interface with DMA channel. (6844)
- ★ Dual timer for real time clock application.
- ★ Powerful on board system monitor (2732). Features commands such as Go To, Alter, Fill, Move, Display, or Test Memory. Also Read and Write Sectors. Boot Normal, Unknown, and General Flex™.

YOUR CHOICE OF POPULAR DISK OPERATING SYSTEMS:

FLEX™ from TSC	$149
OS9™ from Microware	$199

Specify 5-1/4 or 8 Inch

PC BOARD IS DOUBLE SIDED, PLATED THRU SOLDER MASKED, 11 x 11-1/2 IN.

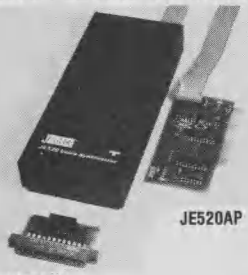
ALL SALES ARE MADE SUBJECT TO THE TERMS OF OUR 90 DAY LIMITED WARRANTY. A FREE COPY IS AVAILABLE UPON REQUEST.

Digital Research Computers
(OF TEXAS)
P.O. BOX 461565 • GARLAND, TEXAS 75046 • (214) 271-3538

TERMS: Shipments will be made approximately 3 to 6 weeks after we receive your order. VISA, MC, cash accepted. Add $4.00 shipping.
USA AND CANADA ONLY

HOME COMPUTER ACCESSORIES

VOICE SYNTHESIZER FOR APPLE AND COMMODORE



JE520AP

JE520CM

- Over 250 word vocabulary-affixes allow the formation of more than 500 words • Built-in amplifier, speaker, volume control, and audio jack • Recreates a clear, natural male voice • Plug-in user ready with documentation and sample software • Case size: 7¼"L x 3¼"W x 1-3/8"H

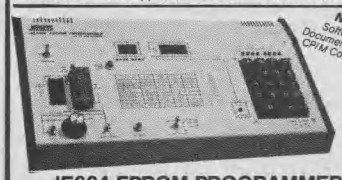
APPLICATIONS: • Security Warning • Telecommunication • Teaching • Handicap Aid • Instrumentation • Games

The JE520 VOICE SYNTHESIZER will plug right into your computer and allow you to enhance almost any application. Utilizing National Semiconductor's DIGITALKER™ Speech Processor IC (with four custom memory chips), the JE520 compresses natural speech into digital memory, including the original inflections and emphases. The result is an extremely clear, natural vocalization.

Coming Soon: **VOCABULARY EXPANSION MODULE**

- Expands to over 1000 basic words
- Allows selection of up to 6 additional word sets

Part No.	Description	Price
JE520CM	For Commodore 64 & VIC-20	$114.95
JE520AP	For Apple II, II+, and IIe	$149.95



JE664 EPROM PROGRAMMER

8K to 64K EPROMS — 24 & 28 Pin Packages

Completely Self-Contained — Requires No Additional Systems for Operation

• Programs and validates EPROMs • Checks for properly erased EPROMs • Emulates PROMs or EPROMs • RS232C Computer Interface for editing and program loading • Loads data into RAM by keyboard • Changes data in RAM by keyboard • Loads RAM from an EPROM • Compares EPROMs for content differences • Copies EPROMs • Power Input: 115VAC, 60Hz, less than 10W power consumption • Enclosure: Color-coordinated, light tan panels with molded end pieces in mocha brown • Size: 15⅜"L x 8¾"D x 3½"H • Weight: 5¾ lbs.

The JE664 EPROM Programmer emulates and programs various 8-Bit Word EPROMs from 8K to 64K-Bit memory capacity. Data can be entered into the JE664's internal 8K x 8-Bit RAM in three ways: (1) from a ROM or EPROM; (2) from an external computer via the optional JE665 RS232C BUS; (3) from its panel keyboard. The JE664's RAMs may be accessed for emulation purposes from the panel's test socket to an external microprocessor. In programming and emulation, the JE664 allows for examination, change and validation of program content. The JE664's RAMs can be programmed quickly to all "1"s (or any value), allowing unused addresses in the EPROM to be programmed later without necessity of "UV" erasing. The JE664 displays DATA and ADDRESS in convenient hexadecimal (alphanumeric) format. A "DISPLAY EPROM DATA" button changes the DATA readout from RAM word to EPROM word and is displayed in both hexadecimal and binary code. The front panel features a convenient operating guide. The JE664 Programmer includes one JM16A Jumper Module (as listed below).

JE664-A EPROM Programmer. $995.00
Assembled & Tested (Includes JM16A Module)

JE665 — RS232C INTERFACE OPTION — The RS232C Interface Option implements computer access to the JE664's RAM. This allows the computer to manipulate, store and transfer EPROM data to and from the JE664. A sample program listing is supplied in MBASIC for CP/M computers. Documentation is provided to adapt the software to other computers with an RS232 port. 9600 Baud, 8-bit word, odd parity and 2 stop bits.

FOR A LIMITED TIME A SAMPLE OF SOFTWARE WRITTEN IN BASIC FOR THE TRS-80® MODEL I, LEVEL II COMPUTER WILL ALSO BE PROVIDED.

JE664-ARS EPROM Prog. w/JE665 Option. $1195.00
Assembled & Tested (Includes JM16A Module)

EPROM JUMPER MODULES — The JE664's JUMPER MODULE (Personality Module) is a plug-in Module that pre-sets the JE664 for the proper programming pulses to the EPROM and configures the EPROM socket connections for that particular EPROM.

JE664 EPROM Jumper Mod. No.	EPROM	Programming Voltage	EPROM MANUFACTURER	PRICE
JM08A	2708	25V	AMD, Motorola, Nat., Intel, TI	$14.95
JM16A	2716, TMS2516 (TI)	25V	Intel, Motorola, Nat., NEC, TI, AMD, Hitachi, Mostek	$14.95
JM16B	TMS2716 (3-V's)	-5V,+5V,+12V	Motorola, TI	$14.95
JM32A	TMS2532	25V	Motorola, TI, Hitachi, OKI	$14.95
JM32B	2732	25V	AMD, Fujitsu, NEC, Hitachi, Intel Mitsubishi, National	$14.95
JM32C	2732A	21V	Fujitsu, Intel	$14.95
JM64A	MCM66764, MCM68L764	21V	Motorola	$14.95
JM64B	2764	21V	Intel, Fairchild, OKI	$14.95
JM64C	TMS2564	25V	TI	$14.95

4-Digit Fluorescent Alarm Clock Kit

- Bright 4-digit 0.5" high display • 10 minute snooze alarm
- AM/PM indicator • Automatic display dimmer

The JE750 Clock Kit is a versatile 12-hour digital clock with 24-hour alarm. The clock has a bright 0.5" high blue-green fluorescent display. The display will automatically dim with changing light conditions. The 24-hour alarm allows the user to disable the alarm and immediately re-enable the alarm to activate 24 hours later. The kit includes all documentation, components, case and wall transformer. Size: 6⅝"L x 3¼"W x 1¾"D.

JE750 Alarm Clock Kit. $29.95

KEYBOARDS — POWER SUPPLIES

217⅛"L x 9.8"W x 3⅛"H

106-Key 8-Bit Serial ASCII Keyboard
• Numeric and cursor keypad • 10 user definable keys • 7 LED function displays • Security lock • N-key rollover • Uses Intel 8048/8748 • Color: white w/black panel • Documentation included • Weight: 6½ lbs.
KB139. . . . . $59.95

197⅛"L x 7⅛"W x 1½"H

Micro-Switch 106-Key Keyboard 8-Bit Serial ASCII
• Numeric and cursor keypad • 8 user definable keys • Japanese/English characters • 8 LED function displays • Documentation incl. • Fits DTE-22 enclosure • Wt: 3½ lbs.
KB106SD29-4. . . . . $29.95

193⅛"L x 6½"W x 1½"H

SMK 103-Key Unencoded Keyboard
Numeric and cursor keypad • SPST mechanical keyswitches • 40-pin header connection • Features: escape, control, cursor keypad, user function and special function keys • Fits DTE-22 enclosure • Weight: 3½ lbs.
KB9000. . . . . $19.95

Power/Mate Corporation +5VDC @ 3 Amp REGULATED POWER SUPPLY
• Input: 105-125/210-250VAC at 47-63 Hz • Output: 5VDC @ 3.0 Amps/6VDC @ 2.5 Amps • Line regulation: 0.05% • Load reg.: 0.1% • Open frame mounts on any 1 of 3 surfaces • Size: 4⅞"L x 4"W x 2⅛"H • Weight: 2 lbs.
EMA5/6B. $29.95

Power/Mate Corporation +5VDC @ 6 Amp REGULATED POWER SUPPLY
• Input: 105-125/210-250VAC at 47-63 Hz • Output: 5V @ 6A/6V @ 5A • Line reg.: 0.05% • Load reg.: 0.1% • Open frame mounts on any one of three surfaces • Size: 5⅝"L x 4⅞"W x 2⅞"H • Wt: 4 lbs.
EMA5/6C. $39.95

Switching Power Supply for APPLE II, II+ & IIe™
• Can drive four floppy disk drives and up to eight expansion cards • Short circuit and overload protection • Fits inside Apple computer • Fully regulated +5V @ 5A, +12V @ 3A, -5V @ .5A, -12V @ .5A • Direct plug-in power cord included • Size: 9⅞"L x 3½"W x 2¼"H • Weight: 2 lbs.
KHP4007. . . . . $79.95

POWER SUPPLY +5VDC @ 7.5 AMP, 12VDC @ 1.5 AMP SWITCHING
Input: 115VAC, 50-60Hz @ 3 amp/230VAC, 50Hz @ 1.6 amp. Fan volt./power supply select switches (115/230VAC). Output: 5VDC @ 7.6 amp, 12VDC @ 1.6 amp. 8 ft. blk. pow. cord. 11½"W x 13¼"D x 3¾"H. Wt. 6 lbs.
Part No. PS94VDS . . . . . $39.95 each

POWER SUPPLY 4-Channel Switching
Microprocessor, mini-computer, terminal, medical equipment and process control applications. Input: 90-130VAC 47-440Hz. Output: +5VDC @ 5A, -5VDC @ 1A, +12VDC @ 1A, -12VDC @ 1A. Line reg.: ±0.2%. Ripple: 30mV p-p. Load reg.: ±1%. Overcurrent protection. Adj. 5V main output ± 10%. 5-3/8"L x 1-7/8"W x 4-15/16"H. Wt. 1½ lbs.
Part No. FCS-604A . . . . . $69.95 each

DISKETTES AND ACCESSORIES

5¼" and 8" Diskettes

SSDD = Single Sided Double Density DSDD = Double Sided Double Density
SSQD = Single Sided Quad Density DSQD = Double Sided Quad Density

ULTRA MAGNETICS — 5¼" DISKETTES

Part No.	Description	Boxed	PRICE
UM51401	5¼" SSDD Soft Sector with Hub Ring and Envelope	10	$24.95
UM00086	5¼" SSDD Soft Sector with Hub Ring (Bulk)	100	169.95
UM52401	5¼" DSDD Soft Sector with Hub Ring and Envelope	10	32.95
UM00140	5¼" DSDD Soft Sector with Hub Ring (Bulk)	100	239.95
UM51801	5¼" SSQD Soft Sector with Envelope (96TPI)	10	34.95
UM00214	5¼" SSQD Soft Sector 96TPI (Bulk)	100	259.95
UM52801	5¼" DSQD Soft Sector with Envelope (96TPI)	10	41.95
UM00217	5¼" DSQD Soft Sector 96TPI (Bulk)	100	299.95

SK (ESKEI) — 5¼" DISKETTES

SK1D	5¼" SSDD Soft Sector with Hub Ring and Envelope	10	20.95
SK1DB	5¼" SSDD Soft Sector with Hub Ring (Bulk)	100	149.95
SK2D	5¼" DSDD Soft Sector with Hub Ring and Envelope	10	26.95
SK2DB	5¼" DSDD Soft Sector with Hub Ring (Bulk)	100	199.95

ULTRA MAGNETICS — 8" DISKETTES

UM81728	8" SSSD IBM-Compatible (128 B/S, 26 Sector) and Envelope	10	34.95
UM00825	8" SSSD IBM-Compatible (128 B/S, 26 Sector) Bulk	100	339.95
UM82726	8" DSDD Soft Sector (Unformatted) with Envelope	10	49.95
UM00840	8" DSDD Soft Sector (Unformatted) Bulk	100	299.95

All diskettes listed above are soft-sectored. All have Hub Rings with the exception of the Quad Density and the 8" Series.

DISKETTE ACCESSORIES

Disk Minder

• Attractive, functional disk storage system • 50 (8") or 75 (5¼") disk storage capacity • Easy filing and retrieving • Protects disk from dust contamination • Molded from durable smoked plastic with front carrying handle • Size: 7"W x 6½"H x 9¼"D • Weight: 2 lbs.

Part No.	Description	Price
DM75	Stores 75 (5¼") Diskettes	$19.95 each
DM50	Stores 50 (8") Diskettes	$29.95 each

MINI-PAK
• Stores 10 (5¼") diskettes • Protects disk from dust contamination • Durable smoked plastic • Size: 6"L x 5⅜"H x 1⅞"D

Part No.	Description	Price
MP-10	Stores 10 (5¼") Diskettes	$2.95 each

Diskette Envelopes

Sleeves for Bulk-Purchased Diskettes
• Tear proof • Lint/contaminant free • Anti-static protect. • Write/read

Part No.	Description	Price
MP5201	10 White 5¼" Envelopes	10 for $ 1.49
MP5201-100	100 White 5¼" Envelopes	100 for $10.95
MP8201	10 White 8" Envelopes	10 for $ 1.95
MP8201-100	100 White 8" Envelopes	100 for $17.95

Vinyl Pages

For 3-Ring Binders

Protects disks from dirt, scratches, dust, spills, and other contaminants. Reduces the risk of getting dirt, near your disk drive head. Size: 8½" x 11".

Part No.	Description	Price
PC001	2 Pocket 5¼" Vinyl Page	10 for $7.95
PC001-2	4 Pocket 5¼" Vinyl Page	10 for $8.95
PC014	1 Pocket 8" Vinyl Page	10 for $7.95
PC014-2	2 Pocket 8" Vinyl Page	10 for $8.95

Mail Pak™

- Holds up to 3 diskettes
- Ideal for mailing and retail packaging
- Dust proof and durable
- Transparent sleeve allows easy identification

Part No.	Description	Price
MP-05	Holds 3 ea. 5¼" Diskettes	$2.49 each
MP-08	Holds 3 ea. 8" Diskettes	$3.95 each

$10.00 Minimum Order — U.S. Funds Only
California Residents Add 6½% Sales Tax
Shipping — Add 5% plus $1.50 insurance
Send S.A.S.E. for Monthly Sales Flyer!

Spec Sheets — 30¢ each
Send $1.00 Postage for your *FREE 1984 JAMECO CATALOG*
Prices Subject to Change

Mail Order Electronics • Worldwide

MasterCard **Jameco ELECTRONICS** VISA®

1355 SHOREWAY ROAD, BELMONT, CA 94002

3/84 ***PHONE ORDERS WELCOME — (415) 592-8097 Telex: 176043***

DISK DRIVES AND CABLE

8" FLOPPY DISK DRIVE

- Single-Sided
- 77 Tracks
- 400/800K Bytes Capacity
- Industry Standard

The **FDD100-8 8" Floppy Disk Drive** (Industry Standard) features single or double density. Recording mode: FM single, MFM double density. Transfer rate: 250K bits/sec. single density; 500K bits/sec. double density. The FDD100-8 is designed to work with the single-sided soft sectored IBM Diskette I, or eq. disk cartridge. Power: 115VAC @ 50-60Hz, + 24VDC @ 1.7 amps max., +5VDC @ 1.2 amps max. Unit as pictured above (does not include case, power supply, or cables). Size: 8.55"W x 14"L x 4.5"H. Weighs 12 lbs. Incl. 96-pg. manual.

FDD100-8 . . $169.95 ea.

5¼" APPLE™ Direct Plug-In Compatible Disk Drive

• Uses Shugart SA390 mechanics • 143K formatted storage • 35 tracks — compatible with Apple controller • Complete with connector and cable — just plug into your disk controller card • Size: 6"L x 3½"W x 8-9/16"D • Weight: 4¼ lbs.
Part No. ADD-514 $195.95

5¼" TEAC DISK DRIVE

Single-Sided Half-Height 5¼" Drive

• Single or double density • 48 TPI • 40 tracks • 6ms track to track • 5W power consumption • Brushless DC direct-drive motor • 160KBytes formatted storage • One year warranty parts and labor

Double your work space with the TEAC 5¼ FLOPPY DISK DRIVE. Because the TEAC FD55A Drive is half the height of conventional drives, you can fit up to four TEAC drives in the same space where two conventional drives fit. Or, have room for two floppy disk drives and a hard disk drive. Includes operating manual. Requires: +5VDC @ .55A and +12VDC @ .3A. Size: 5¾"W x 1⅝"H x 8"D. Wt: 3 lbs.
FD55A. $249.95

5¼" SHUGART DISK DRIVE

Double-Sided Half-Height 5¼" Drive

• Single or double density • 48 TPI • 40 tracks • 409KBytes formatted storage • 6ms track to track • Brushless DC direct-drive motor • One year warranty parts and labor • Media and interface compatible with the SA450

Backed by the industry's largest and most experienced engineering sales and service organization, the SA455 is the smart solution for a wide range of desktop applications. Includes operating manual. Requires: +5VDC @ .7A and +12VDC @ .75A. Size: 5.88"W x 1.63"H x 8"D. Wt: 3.3 lbs.
SA455. . . . . $259.95

5¼" PANASONIC DISK DRIVE

Double-Sided Half-Height 5¼" DRIVE
Shugart SA455 Equivalent

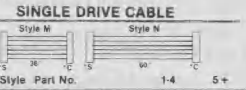
• Single or double density • 48 TPI • 40 tracks • 6ms track to track • 327KBytes formatted storage • One year warranty parts and labor

The JA551 is perfect for word processors, personal and portable computers, small business computers and terminal add-ons. Includes operating manual. Requires: +5VDC @ .9A and +12VDC @ 1 A. Size: 5.88"W x 1.63"H x 8"D. Wt: 3.3 lbs.
JA551-2. . . . . $239.95

SIGNAL CABLES

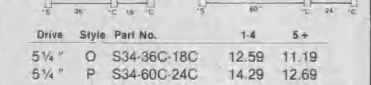
5¼" DRIVES USE 34-PIN ASSEMBLIES
8" DRIVES USE 50-PIN ASSEMBLIES
*S = SOCKET CONNECTOR
*C = CARD-EDGE CONNECTOR

SINGLE DRIVE CABLE



Drive	Style	Part No.	1-4	5+
5¼"	M	S34-36-C	7.95	7.25
5¼"	N	S34-60-C	8.95	8.39
8"	M	S50-36-C	10.95	9.95
8"	N	S50-60-C	12.49	10.95

DUAL DRIVE CABLES



Drive	Style	Part No.	1-4	5+
5¼"	O	S34-36C-18C	12.59	11.19
5¼"	P	S34-60C-24C	14.29	12.69
8"	O	S50-36C-18C	16.49	14.49
8"	P	S50-60C-24C	18.95	16.95

Power Cable Kits

Kits include: Connector shells, connector pins, and power cables.

Power Cable Kit for 5¼" Drive — Price
Part No. PCK-5 . . . . . $2.95

Power Cable Kit for Full-Sized 8" Drive
Part No. PCK-8 . . . . . $3.95

Power Cable Kit for Qume Full-Sized 8" Drive
Part No. PCK-Q $4.95

UV-EPROM Eraser

8 Chips — 51 Minutes

1 Chip — 37 Minutes

Erases 2708, 2716, 2732, 2764, 2516, 2532, 2564. Erases up to 8 chips within 51 minutes (1 chip in 37 minutes). Maintains constant exposure distance of one inch. Special conductive foam liner eliminates static build-up. Built-in safety lock to prevent UV exposure. Compact — only 9.00" x 3.70" x 2.60". Complete with holding tray for 8 chips.

DE-4 UV-EPROM Eraser . . . $79.95
UVS-11EL Replacement Bulb . . $16.95

Peterborough NH 03458

Microcomputing® ● List of Advertisers

Reader Service Number		Page
468	Apple Computer, Inc.	129
403	Automatic Software USA	118
348	Awareco	36
269	BG Micro	14
112	Bill Cole Enterprises	10
8	BRS	17
469	Business Solutions, Inc.	130
407	California Software Products	119
369	Cardco, Inc.	CII
98	Cases, Inc.	125
283	CGRS Microtech	110
170	Chips & Dale	39
473	College Board Publications	130
506	Complexx Systems, Inc.	135
90	Compucover	126
371	CompuServe Info Service	3
401	Computer Creations	118
18	Computer Design Labs	107
488	Computer Designed Systems, Inc.	135
120	Computer Discount of America	123
185	Computer Friends	145
484	Computer Transceiver Systems, Inc.	136
297	Concord Computer Products	123
252	Cuesta Systems	126
23	Data Base Industries	31
*	Data Terminals & Communications	33
319	Dayton Hamvention	139
355	Dennison Computer Supplies, Inc.	CIV
*	Digital Research Computers	112, 113
392	Diskette Connection	106
60	Eastern House Software	106
406	Einstein Corp., The	119
169	Elcomp Publishing, Inc.	63
21	Elek-Tek, Inc.	55
192	Ellis Computing	9
78	Expotek	56
307	Friendly Computer Center	15
481	Gavilan Computer Corp.	135
177	Global Equipment Co.	110
17	Gloucester Computer, Inc.	58
6	H & E Computronics	CIII
236	Heath Co.	81
128	Innovative Technology	126
464	Intelligent Technologies International	129
41	Jameco Electronics	114
*	jr Subscription	59
134	JV Software	126
489	Kalglo Electronics Co., Inc.	134
37	LCD Computer Services	126
463	Lindbergh Systems	129
373	Logical Devices, Inc.	24
204	Lyben Computer Systems	126
465	MAG Software, Inc	128
461	Mark of the Unicorn	128
66	Mastermind Computing	44
316	MFJ Enterprises	45
510	Micro Products Engineering, Inc.	137
471	Micro Software International	130
*	**Microcomputing**	
	*Back Issues	111
	*Dealer Sell	133
	*Foreign Dealers	111
	*Microcomputing Sales Dept.	98
	*Microcomputing Subscriptions	131
	*Moving	123
	*Subscription Problems	126
	*University Micros	126
472	Microsoft Corp.	129
485	MicroStandard Technologies, Inc.	134
477	Midwest Micro, Inc.	130
235	Midwest Micro, Inc.	139
402	Midwest Software Associates	118
409	Modular Software Associates	119
504	Mountain Computer, Inc.	137
312	Mylstar Electronics	132
*	NRI Schools	93
202	Orcin Corp.	86
460	Ovation Technologies	128
212	Pacifica Technology	144
263	Palantir Software	76
480	Panasonic	134
179	Paradigm Consultants	25
511	Polaroid	136
386	Public Domain	10
410	Radio Shack	119
466	Relational Database Systems, Inc.	129
15	Rio Grande Software	95
191	Riverlake Systems	24
405	Roque River Software	118
474	RoseSoft	133
*	RUN Subscription	18
482	STM Electronics	135
475	Scientific Software Products, Inc.	130
408	Sextant Systems	119
496	Smith Corona	137
356	Soft Style	20
271	Softlogic Solutions	71
211	Software Services	86
501	Sony Communications Products Co.	137
479	Spinnaker Software	133
244	Sun Research, Inc.	62
503	Systems Peripherals Consultants	136
295	SWP, Inc.	127
189	Tab Sales	139
493	Teletex Communication Corp.	138
341	TMSI	103
505	TNW Corp.	137
241	Thirdware	11
*	Video Games International	87
491	Vistar Corp.	136
483	Visual Computer, Inc.	134
478	Waveform, Inc.	133
	Wayne Green Inc.	
395	Low Cost Costing	51
*	Manuscript Ad	138
393	Prog Eng/Hob	99
*	Shelf Boxes	138
367	WG Books	91
498	Western Graphtec, Inc.	137
216	Western Wares	126
296	Writing Consultants	86

*This advertiser prefers to be contacted directly.

This index is provided as an additional service. The publisher does not assume any liability for errors or omissions.

For further information from our advertisers, please use the Reader Service card.

CALENDAR

The Microcomputing Jungle—Kansas

The University of Kansas has announced that a two-day conference, The Microcomputing Jungle: Impact on Health Care, will be offered March 1 and 2.

The course will address the problems of computerizing medical offices and the implications of a national information network. University credit is available.

Contact Jan Johnson, Office of Continuing Education, for more information at the University of Kansas Medical Center, 39th and Rainbow, Kansas City, KS 66103; 913-588-4480.

BizCom as Usual in New Hampshire

On March 9–11, Manchester, NH will host BizCom '84, a business equipment and computer show. Business and personal computer vendors, software companies and peripheral marketers will be represented.

Contact Joan Shea for more information at BizCom '84, 115 Front St., Manchester, NH 03102.

H-P Seminars

Hewlett-Packard Co. is sponsoring a series of free-to-the-public, two-day showcases and seminars. Productivity '84 will feature demonstrations and displays that explore ideas and solutions about increased productivity. Twenty-five new products and systems solutions will be highlighted.

Productivity '84 is touring the country this spring. This month, the show will be in Washington, D.C. March 12 and 13 and in Toronto March 19 and 20. April shows are scheduled for Philadelphia, Boston and Teaneck, NJ. Telephone 800-554-4466 to register.

Interface '84 in Las Vegas

The convention center in Las Vegas is the site of this year's Interface '84 conference and exposition on March 12–15. More than 275 computer companies will exhibit at the show; the 54-session conference will address desktop proliferation and AT&T divestiture.

The Industry Awards banquet, black tie optional, will be held March 13 in conjunction with the show.

For registration information, contact The Interface Group at 300 First Ave., Needham, MA 02194; 617-449-6600 or, from outside Massachusetts, 800-325-3330.

Micro/Set in the Motor City

Micro/Set '84, an exposition and conference for scientists, engineers and technicians, is scheduled for March 13–15 in Detroit.

The conference is sponsored by the Engineering Society of Detroit. Selected papers will concentrate on micros and personal computer applications in research, design and manufacturing. A technical program will round out the agenda.

For more information, contact Marla Janess at 313-832-5400.

Telecommunications Down Under

The first Australian Telecommunications Exhibition and Conference will be held March 13–15 in Melbourne, Australia.

Many new telecommunications products will be unveiled at the exhibition; the conference will focus on technological, political and regulatory changes in the field of telecommunications.

For more information, contact Riddell Exhibitions Promotions at 137-141 Burnley St., Richmond, Victoria 3121, Australia; 03-429-6088, or Telex AA149MTC85.

Silicon Valley Fever

The Annenberg School of Communications at the University of Southern California continues its lecture series, Silicon Valley Fever, this month with a talk from MIT's Seymour Papert. On March 14, Papert will discuss "Computing in the Lives of Our Children."

The talk is part of a series exploring the interaction between people and computers. For more information, contact Diane Woods at 213-743-5976.

Simulations in Florida

The seventeenth annual Simulation Symposium is slated for March 14–16 in Tampa, FL. The symposium will feature the latest developments in simulation by digital computer; papers will be presented on a variety of simulation applications.

The show is being sponsored by the IEEE Computer Society, among others. More information is available from A. Kran, IBM Corp., B/300-40E Hopewell Junction, NY 12533; 914-894-7142.

MACUL—Grand Rapids, MI

The Michigan Association for Computer Users in Learning (MACUL) is sponsoring its eighth annual conference March 15 and 16. The conference, at the Grand Plaza Hotel in Grand Rapids, will feature more than 120 sessions focusing on educational computing from early childhood through post-secondary education.

Several featured speakers will address the group. More information can be obtained from Carolyn Gilbreath, Oakland Schools, 2100 Pontiac Lake Road, Pontiac, MI 48054; 313-858-1991.

Micros in Education—AZ

Literacy Plus, the 1984 Microcomputers in Education Conference, will be held March 15 and 16 at the Arizona State University in Tempe, AZ.

Two preconferences are also slated: one addresses general research; the other is titled "Microcomputers and the Writing Process."

Contact Ruth Camuse, College of Education, Payne B 47, Arizona State University, Tempe, AZ 85287; 602-965-7363.

Phoenix Expo

March 16–18 are the dates for this year's Phoenix Computer Showcase Expo at the Phoenix Civic Center. The show is designed to attract business, professional and corporate users of small computer and word processing systems.

In addition to the exhibition, the Small Computer College will offer elementary and advanced seminars at no extra cost. For more information, contact Linda Yogel or Peter Yound, The Interface Group, 300 First Ave., Needham, MA 02194; 617-449-6600 or, from outside Massachusetts, 800-325-3300.

Forth at the West Coast Computer Faire

The ninth West Coast Computer Faire will be held in San Francisco, CA on March 22–25. The trade and end-user exposition will be complemented by a conference featuring more than 100 speakers presenting educational forums.

Additionally, the Forth Interest Group (FIG) will offer workshops, tutorials and speakers addressing various aspects of the Forth computer language.

For more information on FIG and Faire activities, contact PO Box 1105, San Carlos, CA 94070; or call the FIG hotline at 415-962-8653.

SW Computer Conference—Oklahoma City

The seventh annual Southwest Computer Conference is slated for March 27–29. The conference is designed to aid both management and technical personnel.

Over 50 seminars will complement the exhibition. Contact E.Z. Million, SWCC, for more information at PO Box 950, Norman, OK 73070; 405-329-3660.

Miami Forecast—Graphics

The sixth annual Frost and Sullivan conference on computer graphics will be held March 28–30 in Miami. Carl Machover, an expert in the field, will chair the conference, which is titled "Computer Graphics '84: Hardware/Software Assessments and Forecasts."

The conference will also discuss CAD/CAM systems, computer health and safety and graphics hardware. For more information, contact Wendy Engleberg or Fred Rackmil at 212-233-1280.

IBM User's Group Conference—OH

Cincinnati, OH is the site of this year's spring conference of Common, a worldwide IBM user's group. The conference, scheduled for April 1–5, will feature almost 200 presentations covering state-of-the-art IBM topics.

Special management sessions will be presented by Northwestern University. For more information about either Common or the spring conference, contact David Lister, Administrative Director, Common, 435 N. Michigan Ave., Suite 1717, Chicago, IL 60611; 312-644-0828.

DEALER DIRECTORY

Woodbridge, CT

RIP-SOFT! New game spoofs computer bizz. Break the 'calc habit—IBM PC software—See how—Open Basic—Program catalog—Programmer's toolbox—Utilities and fun. **People Systems, Ltd., 78 Maplevale Drive, Woodbridge, CT 06525; 393-3913.**

Aurora, IL

Full line of Apple Computer and Fortune Computer, Hewlett-Packard Personal Computers, Calculators and Supplies. IDS Prism, SMC and Daisywriter Printers. **Farnsworth Computer Center, 1891 North Farnsworth Ave., Aurora, IL 60505 (851-3888) and 383 East North Ave., Villa Park, IL 60181 (833-7100).**

Nokomis, FL

We are the leading area computer store. We carry Cromemco, Apple, Vector Graphic; printers and terminals. We offer full software support including G/L, A/R, payroll and word processing. **Computer Centre, 909 S. Tamiami Trail, PO Box 130, Nokomis, FL 33555. 484-0421.**

Dealers: Listings are $15 per month in prepaid quarterly payments, or one yearly payment of $150, also prepaid. Ads include 25 words describing your products and services plus your company name, address and phone. (No area codes or merchandise prices, please.) Call Marcia at 603-924-9471 or write *Microcomputing*, Ad Department, Peterborough, NH 03458.

CLASSIFIEDS

Classified advertisements are intended for use by persons desiring to buy, sell or trade used computer equipment. No commercial ads are accepted.

Two sizes of ads are available. The $5 box allows up to 5 lines of about 35 characters per line, including spaces and punctuation. The $10 box allows up to 10 lines. Minimize use of capital letters to save space. No special layouts allowed. Payment is required in advance with ad copy. We cannot bill or accept credit.

Advertising text and payment must reach us 60 days in advance of publication (i.e., copy for March issue, mailed in February, must be here by Jan. 1). The publisher reserves the right to refuse questionable or inapplicable advertisements. Mail copy with payment to **Classified *Microcomputing*, Peterborough, NH 03458.** Do not include any other material with your ads as it may be delayed.

For Sale: Used Wire-wrap tools: 115v gun, $60. OK hobby-wrap with Nicads, $20. Vector slit-n-wrap, $17. OK just-wrap, $8. Bob Levine, 32 King St., NY 10014; 212-691-2897.

The Egyptian Management Information Center, 93 Kasr Eleiny St., Cairo, Egypt, phone 27525, telex 92185, Hapi Un, is looking for interested manufacturers to market their hardware and software in Egypt and the Middle East. Contact Adel Fahmy, Ph.D.

LOWEST POSSIBLE PRICES ANYWHERE!!
Computers (Hardware-Software-Printers)
AUDIO (Most Major Brands)
Video Recorders-Car Stereo-Televisions
Major Appliances-ALL NEW-Fully Warranteed!
MC/VISA - ELECTRIFIED DISCOUNTERS
996 Orange Ave., West Haven, CT 06516
203-937-0108

Wanted—Do you have a PC? Earn a guaranteed $240 to $480 weekly while using your PC. No experience or skills required. For details, send a stamped self-addressed envelope to B. Harley, 1104 Stanley Ave., Columbus, OH 43206.

CLUB NOTES

Colorado DACS

The Denver Amateur Computer Society, DACS, is a club with broad computer interest. They meet for a general meeting on the third Wednesday of each month at 7 p.m. (a beginner's session starts at 6:30 p.m.) at the Pomona Senior High School, 8101 Pomona Drive, Wheatridge, CO 80034.

DACS offers a variety of special interest groups (e.g., CP/M, Club Computer, TRS-80, Denver Osborne Group, Boulder Osborne User's Group and IBM PC). Each member receives a copy of the DACS monthly newsletter.

For information, contact Connie Ulehla, PO Box 477, Wheatridge, CO 80034; 303-424-8540.

San Carlos FIGs

The Forth Interest Group, FIG, is a nonprofit world-wide organization of more than 3800 members devoted to the Forth computer language. The group sponsors a FIG-Tree (an on-line, free and totally interactive conference tree). Branches include: Dialog, Calendar of Events, Want Ads, Employment Opportunities, Forth Ethics, Forth Standards, Humor, Users and Vendors.

In addition to providing the FIG-Tree hotline, a subscription to the group's bimonthly publication is provided to all members.

For details, call FIG Hot-line at 415-962-8653 or write FIG, PO Box 1105, San Carlos, CA 94070.

Long Island Computer Group

The Long Island Computer Association, Inc., (LICA) is open to amateurs and professionals interested in computers, applications, programming and related subjects.

Dues are $12 per year. Members receive a subscription to the group's monthly publication. They can place noncommercial and commercial ads in the newsletter.

LICA meets at 8 p.m. on the third Friday of each month in the New York Institute of Technology, Old Westbury Campus, Building 500, Room 508.

For information, contact Al Levy, PO Box 71, Hicksville, NY 11802; 516-293-8368.

North Texas IBM PC Group

The North Texas IBM Personal Users' Group is an independent, nonprofit group that isn't associated with IBM Corp. The group meets at 9:30 a.m. on the second Saturday of each month at the Jesuit College Preparatory School, 12345 Inwood Road, Dallas, TX.

Group members receive free newsletter subscriptions where they can place advertisements. Those who submit original articles are offered a complimentary membership to the group.

For information, call Alan Elliot, 214-941-8475.

Micro Software Digest

Compiled by Tracy Mayor

Micro Software Digest presents capsulized software reviews from various computer-related publications.

The Ultimate

System Requirements: IBM PC; DOS 1.1 or 2.0; 64Kb; at least one disk drive
Manufacturer: Computer Creations, 766 El Camino Real, San Carlos, CA 94070
Price: $385
Comments: The latest entry into the word processor field is The Ultimate—and according to the review, this program nearly is the ultimate, with the exception of a few flaws. The product combines a powerful text editor with easily executed commands, a typehead buffer, a spelling checker, a database manager, a mailing list manager and spreadsheet capabilities.

The Ultimate is divided into four modules: Base, which is the database manager, Word, Mail and Spell. Although the program suffers from a "lack of finesse," the review opines that the word processor still functions in a "likable manner." Reader Service number 401.

(Reviewed in Softalk for the IBM Personal Computer, *December 1983)*

Write Away

System Requirements: Apple II or II Plus; 48Kb RAM; ROM Applesoft; at least one disk drive
Manufacturer: Midwest Software Associates, Box 301, Saint Ann, MO 63074
Price: $175
Comments: The review observes that, despite the overcrowding in the Apple word processing market, Write Away "has capabilities that should distinguish it from the pack."

With Write Away, it's possible to manage a mailing list, interface with DIF files and use a sophisticated macro command scheme. You can review your text a letter, a word or a page at a time. You can also search through your document for a particular word or phrase, which can then be easily replaced throughout the text using a single command.

The processor uses Apple text files to store documents, so you can use it to create EXEC files or to edit Applesoft programs—a boon to programmers.

The program isn't copy protected and is "delightfully" error-trapped.

The review concludes, "There are so many positive features incorporated into Write Away that it would be hard not to recommend it for use in business, or by any serious student or professional writer." Reader Service number 402.

(Reviewed in Softalk, *October 1983)*

Autodex

System Requirements: 8080- or Z80-based system or IBM PC; CP/M or PC DOS; 48Kb RAM; one disk drive, either floppy disk or hard disk.
Manufacturer: Automatic Software USA, 1035 Santa Barbara St., Santa Barbara, CA 93101
Price: $150
Comments: If you've been having trouble keeping your CP/M files manageable, Autodex can help to bail you out. The program offers a comprehensive set of one-stroke commands for performing a variety of common operations, including rename, erase, copy, list, view and execute.

Autodex lets you sort files by name, type or size. The display screen is divided into two parts: the disk area reveals disk information and commands on the top three rows; the file area (the rest of the screen) lists files and identifies type, size, name, date of last change and description. In all, there are ten disks commands and 13 file commands.

Autodex has to be installed on your system but, according to the review, "the process has been simplified through use of menus." Reader Service number 403.

(Reviewed in InfoWorld, *July 25, 1983)*

SPF/PC

System Requirements: IBM PC; DOS 1.1 or 2.0; 128Kb; at least one disk drive
Manufacturer: Rogue River Software, 2822 Tahitian Ave., Medford, OR 97504
Price: $149.95
Comments: SPF/PC is a full-screen data editor modeled after ISPF, IBM's mainframe editor. The package should be particularly useful to businesses wishing to interact micros and mainframes.

SPF/PC supports almost all of its big brother's commands; it also boasts a few of its own. Transfer and Check, two new additions, "should be extremely useful" according to the review.

The program offers four-way scrolling—horizontal scrolling lets you edit data lines of up to 240 characters. SPF/PC has split- or dual-screen capabilities as well, so you can work on more than one file at a time. Also of interest to businesses is the fact that screen colors are closely akin to those of the 3279 mainframe terminal.

The review also points out that the software company is well-known for its user support. This editor "really makes editing data a pleasure." Reader Service number 405.

(Reviewed in Softalk for the IBM Personal Computer, *October 1983)*

The Einstein Compiler

System Requirements: Apple II or II Plus; 48Kb RAM; ROM Applesoft; at least one disk drive
Manufacturer: The Einstein Corp., 11340 West Olympic Blvd., Los Angeles, CA 90064
Price: $119.95
Comments: Taking the various requirements of a good Basic compiler into consideration, the review enthuses that "Einstein is a winner in all categories."

Einstein "shows a significant improvement over the competition" in terms of space requirements. Einstein fairly consistently creates code that is less than twice the length of the original program—an improvement over other compilers.

The compiler is easy to use and comes with excellent documentation, according to the review. Options are included that allow you to compress code, disable a line trace and disallow unstructured for...next loops. The program is designed so that compiler directives in the form of REM statements usually aren't necessary.

The review sums up that "It's hard to find fault with the Einstein Compiler." Reader Service number 406.

(Reviewed in Softalk, *May 1983)*

California 10 Pak

System Requirements: IBM PC; 64Kb; one or two disk drives
Manufacturer: California Software Products, 525 North Cabrillo Park Drive, Santa Ana, CA 92701
Price: $100
Comments: As the review points out, "For all the software written for the IBM PC, there is still a woeful shortage of utility programs." California 10 Pak is a package of, yes, ten utility programs designed to fill that gap.

As with most utility packages, some programs are merely useful and a few are outstanding. Three of the most powerful programs are Disasm, Ccomp and Cdiff. Disasm lets you disassemble COM or EXE files or areas of memory. The program uses labels for reference calls, jumps and moves so that changes can be made to the file. Disasm is "indispensable" if you want to discover how a particular program is put together or if you want to learn 8088 assembly language.

Ccomp compares two files and shows their differences. Unlike the DOS Comp command, this one shows all the dissimilarities, not just the first ten. Cdiff performs the same function for text files and can look ahead as many as 100 lines for insertions and deletions, so the comparison is a sound one.

Data from all utilities can be sent to the printer. The review recommends that "California 10 Pak is a welcome addition." Reader Service number 407.

(Reviewed in Softalk for the IBM Personal Computer, *August 1983)*

Compac

System Requirements: IBM PC; 64Kb RAM; one or two disk drives
Manufacturer: Sextant Systems, Box 251, Holmdel, NJ 07733
Price: $49
Comments: Compac is designed to shrink files so that they take up less space on your disks and load faster. The program not only cuts ASCII-coded files down, but shrinks command files as well.

As the review points out, "compacted ASCII programs are good for archival purposes, but they can't be read by the program that uses them." Consequently, a companion program, Decompac, is included to unshrink your compacted files if and when you need them back.

If you're looking to make the most of your disk space but don't dare throw away some of those less used programs, Compac forces files "to go on a diet and make the most of disk space." Reader Service number 408.

(Reviewed in Softalk for the IBM Personal Computer, *September 1983)*

Newbasic 2.0

System Requirements: TRS-80 Model I or III; 48Kb; two disk drives
Manufacturer: Modular Software Associates, 209 18th St., Huntington Beach, CA 92648
Price: $39.95
Comments: If graphics are your gripe about your TRS-80, Newbasic 2.0 may help solve your problems. The program adds several color computer-like commands to help graphics programmers.

Newbasic 2.0, which operates under DOS PLUS, NEW DOS, LDOS, or TRS DOS, adds 50 new commands to the existing Basic commands. Highlights include: Line, which allows you to instantly draw a line from any point to any other point; Circle, which will draw a circle, ellipse or arc after you specify a starting point and a radius; and Draw, which the review opines is "probably the most versatile of the graphics commands."

Sound effects are also much simpler with Newbasic 2.0. With an additional amplifier, you only need to indicate a frequency and duration to get your computer singing.

Other benefits of the program include a spooler/despooler so you can continue work on your files while printing, a new trace facility, enhanced error detection and an Unnew command, which prevents fatal errors. The review recommends, "If you use graphics and find Disk Basic lacking in commands, Newbasic 2.0 is definitely for you." Reader Service number 409.

(Reviewed in Softside, *#45)*

Disk Trendex

System Requirements: TRS-80 Model I or III
Manufacturer: Radio Shack, 1800 One Tandy Center, Fort Worth, TX 76113
Price: $49.95
Comments: Election year stock markets are traditionally active, and your TRS-80 may help you sniff out a few good deals. Disk Trendex will help you keep track of stock market statistics and your own portfolio. Of course, predicting the vagaries of Wall Street is an impossibility, but this program can help you keep organized by producing clear information on what's happening to which stock.

The program offers a choice of four analysis tracks: short-term market trend, intermediate-term market trend, long-term market trend and intermediate-term stock portfolio.

Short-term covers a two- to ten-week period; long-term gives you a year at a glance. To start the program, you need 28 day's worth of information. For those of you who don't have quite that extensive a newspaper collection, Trendex Research Corp. will provide start-up data for a small fee.

The intermediate and long-term portions use the New York Stock Exchange Index. For your personal portfolio analysis, the program even has a provision to account for split stocks, should you be so lucky. Reader Service number 410.

(Reviewed in Personal Computing, *November 1983)*

InfoWorld, published by Popular Computing, Inc., 375 Cochituate Road, Box 880, Framingham, MA 01701.

Popular Computing, published by BYTE Publications, Inc., 70 Main St., Peterborough, NH 03458.

SoftSide, 6 South St., Milford, NH 03055.

Softalk and *Softalk for the IBM Personal Computer*, 11160 McCormick St., North Hollywood, CA 91601.

Table. Addresses for the magazines publishing the software reviews digested in this department.

Listing 3. LightTrace—a fast two-player graphics game written in Apple Pascal 1.1.

```
VAR X1,Y1,X2,Y2,           (* LOCATION OF LIGHTTRACES *)
    ANG1,ANG2,             (* HEADINGS OF LIGHTTRACES *)
        SPD,               (* SPEED FACTOR *)
          I,               (* INDEX VAR *)
       COLL:INTEGER;       (* COLLISION FLAG *)
      DUMMY:CHAR;          (* DUMMY CHARACTER *)
  BSCORE,OSCORE:INTEGER;   (* PLAYER SCORES *)
  (*$I-*)
  PROCEDURE READI(VAR I:INTEGER);
  (*                                                          *)
  (*     CRASH-PROOF ROUTINE TO INPUT AN INTEGER              *)
  (*                                                          *)

  VAR DUMMY:CHAR;
  BEGIN
    READ(I);
    READ(DUMMY);
    WHILE (IORESULT>Ø) OR (DUMMY<>' ') DO BEGIN
      WRITELN;
      WRITE(CHR(7),'BAD FORMAT-REENTER: ');
      READ(I);
      READ(DUMMY)
    END
  END;
  FUNCTION CLRCRS(X,Y,ANG:INTEGER):INTEGER;
  (*                                                          *)
  (*    THIS FUNCTION TESTS TO SEE IF THE LIGHTTRACE HAS A CLEAR  *)
  (*    PATH OR IF IT IS GOING TO COLLIDE WITH A WALL OR TRACE.   *)
  (*    IT RETURNS O IF THE COURSE IS CLEAR.  1 IN CASE OF        *)
  (*    ANY COLLISION.                                        *)
  (*                                                          *)
  (*    INPUT PARAMETERS:                                     *)
  (*      X,Y,ANG - THE LOCATION AND HEADING OF THE TRACE     *)

   VAR D1,D2,XO: INTEGER;
   BEGIN
     CLRCRS:=Ø;
     (*  SET TESTING PARAMETERS BASED ON COLOR OF TRACE  *)
     IF ODD(X) THEN BEGIN
       D1:=3;
       D2:=5;
       XO:=-1
       END
     ELSE BEGIN
       D1:=5;
       D2:=3;
       XO:=1
       END;
    CASE ANG OF
      Ø:  IF (SCREENBIT(X+4,Y)) OR (SCREENBIT(X+D1,Y)) THEN CLRCRS:=1;
      9Ø: IF (SCREENBIT(X,Y+4)) OR (SCREENBIT(X+XO,Y+4)) THEN CLRCRS:=1;
      18Ø:IF (SCREENBIT(X-4,Y)) OR (SCREENBIT(X-D2,Y)) THEN CLRCRS:=1;
      27Ø:IF (SCREENBIT(X,Y-4)) OR (SCREENBIT(X+XO,Y-4)) THEN CLRCRS:=1
          END
    END;

  PROCEDURE MOVETRACE(VAR X,Y:INTEGER;ANG: INTEGER);
  (*                                                          *)
  (*    MOVES LIGHTTRACE ACROSS GRID ONE UNIT.                *)
  (*                                                          *)
  (*    INPUT PARAMETER:                                      *)
  (*      ANG - CURRENT HEADING                               *)
  (*                                                          *)
  (*    INPUT/OUTPUT PARAMETERS:                              *)
  (*      X,Y - COORDINATES OF LIGHTTRACE, ARE UPDATED AFTER MOVE  *)
  (*                                                          *)
  BEGIN
    PENCOLOR(NONE);
    MOVETO(X,Y);
    TURNTO(ANG);

    (* BLUE TRACE IN EVEN ROWS, ORANGE IN ODD ROWS *)
     IF ODD(X) THEN PENCOLOR(ORANGE)
     ELSE PENCOLOR(BLUE);

     MOVE(4);
     X:=TURTLEX;
     Y:=TURTLEY
     END;
   PROCEDURE TURNTRACE(VAR ANG1,ANG2:INTEGER);
  (*    SCANS KEYBOARD TO SEE IF A MOVEMENT COMMAND HAS BEEN      *)
  (*    ENTERED BY EITHER PLAYER.                             *)
  (*                                                          *)
  (*    OUTPUT PARAMETERS:                                    *)
  (*      ANG1,ANG2 - HEADINGS FOR LIGHTTRACES: ARE UPDATED IF    *)
  (*        COURSE CHANGE HAS BEEN ENTERED                    *)
  (*                                                          *)

  VAR KEY: CHAR;   (* KEYBOARD CHARACTER *)
  BEGIN
    IF KEYPRESS THEN BEGIN
      READ(KEYBOARD,KEY);
      IF KEY IN ['I','J','L','M'] THEN
      CASE KEY OF
        'I': ANG1:=9Ø;
        'J': ANG1:=18Ø;
        'L': ANG1:=Ø;
        'M': ANG1:=27Ø
        END
    ELSE IF KEY IN ['W','A','D','Z'] THEN
      CASE KEY OF
        'W': ANG2:=9Ø;
        'A': ANG2:=18Ø;
        'D': ANG2:=Ø;
        'Z': ANG2:=27Ø
        END
      END
    END;
      PROCEDURE INSTRUCT;
   (*
   (*    PRINTS OUT GAME INTRODUCTION AND INSTRUCTIONS.
   (*
  BEGIN
    PAGE(OUTPUT);
    WRITELN('LIGHTTRACE');
    WRITELN;
    WRITELN('WELCOME USERS.');
    WRITELN;
    WRITELN('  YOUR OBJECT IS TO MANEUVER YOUR LIGHT-');
    WRITELN('TRACE USING THE KEYS SHOWN, FORCING THE');
    WRITELN('OTHER PLAYER TO RUN INTO A WALL OR');
    WRITELN('ANOTHER LIGHTTRACE.');
    WRITELN;
    WRITELN('  GOOD LUCK.');
    WRITELN;
    WRITELN('****************************************')
    END;
```

LETTERS

continued from page 39.

More →

Listing 3 continued.

```
PROCEDURE REZZ;
(*
(*      PRODUCES SOUND EFFECTS AT START.
(*
VAR I,J: INTEGER;

BEGIN
  FOR I := 1 TO 3 DO
    FOR J := 1ø TO 25 DO NOTE(J,3)
END;

PROCEDURE DEREZZ;
(*
(*    PRODUCES CRASH SOUND AT END.
(*

VAR I:INTEGER;

BEGIN
  FOR I := 3ø DOWNTO 5 DO NOTE(I,3);
  NOTE(4,75);
  (* PAUSE *)
  NOTE(ø,2øø)
END;
(* MAIN PROGRAM *)
BEGIN

  (* INITIALIZE SCORES AND GIVE INSTRUCTIONS *)
  BSCORE:=ø;
  OSCORE:=ø;
  INSTRUCT;

(* BEGIN GAME LOOP *)
REPEAT
  WRITELN;
  WRITELN('LIGHTTRACE CONTROL CODES:');
  WRITELN;
  WRITELN('           BLUE    ORANGE');
  WRITELN;
  WRITELN('    UP        I         W');
  WRITELN('  DOWN        M         Z');
  WRITELN(' RIGHT        L         D');
  WRITELN('  LEFT        J         A');
  WRITELN;
  WRITELN;
  WRITE('PLEASE ENTER SPEED (HIGHER #S SLOWER)');
  READI(SPD);
  INITTURTLE;
  MOVETO(ø,ø);
  PENCOLOR(WHITE);
  MOVETO(277,ø); MOVETO(277,189);
  MOVETO(ø,189); MOVETO(ø,1);
  MOVETO(276,1); MOVETO(276,188);
  MOVETO(1,188); MOVETO(1,1);

  (* INITIALIZE LIGHTTRACE POSITION *)
 PENCOLOR(NONE);
 (* EVEN X COORD FOR BLUE TRACE, ODD FOR ORANGE *)
 ANG1:=9ø; X1:=14ø; Y1:=12;
 ANG2:=27ø; X2:=141; Y2:=176;
 COLL:=ø;

 (* WAIT FOR KEYPRESS TO SIGNAL START *)
 READ(DUMMY);
 REZZ;
```

More

```
 (* MAIN LOOP *)
 WHILE COLL=ø DO BEGIN
   (* TIME DELAY FOR SPEED FACTOR *)
   FOR I := 1 TO SPD DO TURNTRACE(ANG1,ANG2);

   (* CHECK FOR NEW DIRECTION, COLLISION, MOVE TRACE *)
     TURNTRACE(ANG1,ANG2);
     COLL:=CLRCRS(X1,Y1,ANG1);
     MOVETRACE(X1,Y1,ANG1);
     TURNTRACE(ANG1,ANG2);
     COLL:=COLL+2*CLRCRS(X2,Y2,ANG2);
     MOVETRACE(X2,Y2,ANG2);
     END;

 (* END OF MAIN LOOP *)
 DEREZZ;
 PAGE(OUTPUT);
 TEXTMODE;

 (* GIVE RESULTS, UPDATE SCORES *)
CASE COLL OF

     1: BEGIN
          GOTOXY(1ø,1ø);
          WRITELN('BLUE TRACE WAS DE-REZZED');
          OSCORE:=OSCORE+1;
        END;

     2: BEGIN
          GOTOXY(1ø,1ø);
          WRITELN('ORANGE TRACE WAS DE-REZZED');
          BSCORE:=BSCORE+1;
        END;

     3: BEGIN
          GOTOXY(1ø,1ø);
          WRITELN('BOTH TRACES WERE DE-REZZED');
        END;
  END;

  GOTOXY(1ø,12);
  WRITELN('DO YOU WISH TO PLAY AGAIN?');
  NOTE(2ø,1øø);
  (* READ LEFTOVER CHARACTERS FROM BUFFER *)
  WHILE KEYPRESS DO READ(KEYBOARD,DUMMY);
     (* NOW READ RESPONSE *)
     READ(DUMMY);
     PAGE(OUTPUT);
     UNTIL DUMMY='N';

   (* END GAME LOOP *)

   (* REPORT RESULTS *)
   GOTOXY(1ø,5);
   WRITELN('RESULTS OF GAME:');
   WRITELN;
   WRITELN;
   WRITELN('BLUE TRACE:   ',BSCORE,' VICTORIES.');
   WRITELN;
   WRITELN('ORANGE TRACE: ',OSCORE,' VICTORIES.');
   WRITELN;
   WRITELN
 END.
```

Delving into the Depths of the 8086/8088 What's the DIF? A Look at Programming with Graphics

The 8086 Book
Russell Rector and George Alexy
Osborne/McGraw-Hill, 1983
2600 Tenth St.,
Berkeley, CA 94710
Softcover, 624 pp., $16.99

The 8086 and 8088 microprocessors, both covered in *The 8086 Book*, are part of the latest generation of CPU chips to appear on the personal computer market.

The 8088 is the chip IBM implemented in the PC. The novice PC user may be unaware of the presence of the 8088 since its operation is masked by IBM's DOS and Basic. Because the system software responds to words as commands, the operator never needs to know the specifics of 8088 machine code.

However, the versatility and power of the IBM PC is a consequence of the 8088's complexity. This complexity involves both programming and electronic circuitry.

Proficiency in dealing with these factors requires a substantial understanding of the microprocessor and its electronic environment. When questions arise, a good source of technical information becomes a valuable tool.

The 8086 Book fulfills the criteria of a good reference text on 8086/8088 operation. Although the PC uses the 8088 processor, the characteristics of the 8086 and 8088 are so similar that the differences can be explained in only four of the 600 pages of text. The most conspicuous difference is the size of the data bus; the 8088 uses only eight data lines while the 8086 uses 16.

Those Computer Melodies

The book is a comprehensive presentation of assembly language concepts and electric design for the 8086/8088. Assembly language programming concerns the direct manipulation of the microprocessor's registers, which are on-chip storage units for information.

To understand the difference between programming in Basic and programming in assembly language, an analogy can be made with a player piano. If the keys of the piano represent the storage registers, assembly language programming is akin to pounding out a tune on the keys.

In contrast, programming in Basic with its English commands is like pushing the pedals while the piano roll determines which keys are depressed.

The 8086 Book was written to provide information on the specifics of CPU operation and control. This is a book for the electronic engineer faced with the task of designing a 16-bit microcomputer and its peripherals, or for the systems software programmer who writes the software to create, for example, higher-level languages such as Basic.

PC computer enthusiasts well-versed in assembly language programming will also want to add this book to their library.

The book is organized into nine chapters. The first six chapters deal with assembly language; the remaining three are on electronic engineering.

Diagrams of the CPU's registers are used throughout the chapters on the instruction set. These diagrams, coupled with assembly language programming examples, satisfactorily convey the function of each instruction.

The appendix contains two lists of the instruction-set mnemonics, the engineering specifications for many of the 8086/8088 support integrated circuits and a discussion of the differences between the 8086 and the 8088. Schematics of circuit design appear throughout the hardware chapters, as do timing diagrams and pin assignment charts. There's no glossary but the index seems adequate for a reference source.

Delight

The dominant topic of the book is the instruction set (or commands) of the 8086/8088 chip and its use. More than half of the text is dedicated to illustrating the commands and their meanings.

I was delighted to find new commands that prove to be real enhancements to the earlier generation of CPU chips. Two interesting commands introduced are the multiplication and division instructions.

It is startling to realize, considering the popularity of small calculators, that no commands existed at the assembly language level on the eight-bit microprocessors for multiplying or dividing. The eight-bit CPUs depended upon complicated software to perform these tasks. A significant simplification of programming results from the inclusion of the multiply and divide instructions on the 8086/8088.

When assembly language programmers learn the instructions available for any CPU, they generally categorize the commands with respect to function. Authors Russell Rector and George Alexy provide a chapter that classifies the instruction set according to usage, breaking the command set down into eight segments, such as data movement, arithmetic and logical instructions. This categorization, and the accompanying program examples, facilitates recall of the extensive command set.

The book is not entirely dedicated to 8086/8088 programming. The hardware and electronics surrounding the CPU are discussed in a fair amount of detail. This detail is necessary for software programming on the assembly level.

The functional aspects of registers, memory, timing and device control are factors the assembly language programmer must consider. Assembly language programming is like driving a car with a stick shift. In a car, the driver must coordinate the clutch, motor speed and gear shift. With assembly language programming, registers, memory and other hardware must be coordinated to achieve the programmer's goal.

Specifics about computer design are valuable in a reference source of this type. However, a computer hobbyist, less schooled in hardware, may be left behind by the book's hardware discussion. The

Circle 120 on Reader Service card.

Circle 297 on Reader Service card.

presentations of 8086/8088 multibus design or multiprocessor configurations require more electronics background than can be taught in one chapter.

Chapter 8, "Basic 8086 Design Single CPU," gives insight into the inner workings of an IBM PC. Elementary components of computer systems and integrated circuits are described and isolated into small blocks. One point of interest is the discussion of operating modes.

The authors instruct that the processor can be "wired" to function in a single CPU environment, called the minimum mode, or in a coprocessor configuration referred to as the maximum mode. The modes differ because of an increased complexity of the control bus, which are the signal lines that inform the CPU the status of the system.

Above Average

The 8086 Book rates above average on subject content. If a question on CPU operation comes up, the answer is covered in some portion of the text. The sequence of chapters, however, leaves something to be desired. The first two chapters are elementary discussions of computer programming, including a basic explanation of a sort routine, a program commonly used to put data in sequence.

The carryover of useful information from these chapters to the rest of the text is not clear. They don't really school the reader in background needed to understand later topics.

Even more curious is Chapter 5, which covers software development. It follows extensive presentations on assembly language programming and yet the content is another elementary tutorial of what editors, assemblers and debuggers are. Anyone able to handle the earlier, highly technical chapters is likely to be familiar with such programs.

What should have been presented is an explanation of the assembler used to create the examples in the book. In more than one instance, the reader's understanding is blocked not by an 8086/8088 concept but by assembler syntax and the quirks of the assembler the authors employed.

An attentive reader of *The 8086 Book* can easily be overwhelmed by its scope. But because many questions arise when assembling a program, programming on the IBM PC requires such a text. Although the PC can perform an astounding amount of work, understanding and controlling it can be quite demanding at the assembly level.

As PC owners become accustomed to the intricacies of their machines and begin expanding the PC's applications, they will become more deeply involved with the details of the 8088. As this occurs and their questions multiply, they will do well to remember that the answers lie in *The 8086 Book*.

Gary Grout
Poulsbo, WA

If you want to keep up with one of the more important areas of microcomputing—the transfer of data between software packages—you'll find this book more than worthwhile.

The DIF File: For Users of VisiCalc And Other Software

Donald Beil
Reston, 1983
11480 Sunset Hills Road
Reston, VA 22090
Softcover, 235 pp., $14.95
Hardcover, $19.95

The Data Interchange Format (DIF) was created to exchange (primarily) numerical data between computer programs. Since it comes from the folks who brought us VisiCalc, this most popular program also serves as a basis for much of the focus of Donald Beil's book, *The DIF File*.

But DIF does much more than allow us to send VisiCalc data to other programs and vice versa. Inside the program itself, the DIF capability gives the VisiCalc user extra power.

When you save a VisiCalc worksheet, you save the entire file. When you pull the worksheet off disk, you reload the entire sheet. With DIF, you can save just a *section* of a worksheet. When you load a DIF file back into memory, you can place it anywhere on the displayed worksheet. You can save a DIF file in a column format (up and down), or if you want to, you can reload it in a row format.

There are a number of ideas on how to use DIF within the VisiCalc environment.

What if you enter huge amounts of data in the wrong areas on a worksheet? You can save it and reload it in the format and locations you intended it to be.

Would you like to consolidate data from several worksheets? You can save the pieces you need as DIF files and then create a new worksheet with exactly the information you want on it, where you need it to be. DIF can often save memory space and even reduce disk access time.

DIF-ficulties

Beil details not only how to do such things in a step-by-step way but also discusses the limitations of the DIF format. For instance, formulas cannot be saved; you can only work with the numerical values those formulas produce.

Once he's given you a basic introduction to the concept of the Data Interchange Format, Beil goes through a number of what he calls Case Studies, with specific examples to show you how to transfer information from one popular program to another. All of the information is presented with a combination of text and visual materials to make it more understandable.

The logical place to start is, of course, with VisiCalc's data transfer to VisiTrend/Plot. You then learn how to transfer information in the other direction—how to create it within VisiTrend/Plot, send it to VisiCalc, manipulate it there and finally return it home to VisiTrend/Plot, where you create graphics based on the modified data.

You also get to "work with" a group of other Apple and IBM PC programs. You see how to send information from VisiCalc to pfs:Graph. You learn how to share data between DB Master and the Executive Secretary, where you can pull information from the file-handling system (DB Master) to use in form letters you create with the word processor (Executive Secretary). There's a utility program called LoadCalc that changes DIF files into text files so they can be used with other packages, and you learn how to use it as an interface between VisiWord (a new word processor from the VisiCalc people) and VisiCalc.

Would you like to pull data from the online database CompuServe and manipulate it with VisiCalc? It's covered.

Do you use Lotus' 1-2-3 on your IBM PC? While an integrated program like 1-2-3 is the wave of the future, unless all of the data you might want to use is in your 1-2-3 files, one day you'll need the DIF format to access and use information from other sources.

If 1-2-3 doesn't contain some function you'd like to perform on your data, it'd be helpful to send it to another program. Beil shows you how.

Have you just started with TK!Solver, but all of your information is collected elsewhere? If it's handled with a program (like VisiCalc) that will create DIF files, you can transfer them to TK!Solver's worksheets.

One of the more interesting parts of all this is that you get an inside look at all of these systems. Naturally, you're not taught how to use every program but, rather, you get a general overview of each package and how it performs. You get to peek at things you might not otherwise know about.

Simplifying DIF

Beil's writing is clear and thorough;

while he makes the process of learning how to use DIF files understandable, you still have to read and comprehend the material. Some of the information might not be helpful to you right now (perhaps you don't use the Executive Secretary or pfs:Graph), but once you know and understand how the DIF process works, this book will help you grow as you do acquire this program or that in the future.

There are some limitations to the DIF process, and every program doesn't handle things in exactly the same way.

As Beil notes, "Be ready to prepare multiple DIF files to make the same data transportable to multiple programs." He offers information on how you should document these transfer files and probably more detail than what you want on how DIF files work inside your microcomputer. Even all this information is covered in a tutorial manner, though, which makes it easier to learn.

The DIF File contains an appendix that lists the technical specifications according to the DIF Clearinghouse. It's a copy of the typewritten definitions for the format and includes two Basic programs—one that creates a DIF file and another that reads it.

There's an alphabetical listing of 82 software programs and the companies whose products use the Data Interchange Format. Finally, there's an index and excellent bibliography.

If you use VisiCalc, buy this book. Even if you don't want to send data to another program, you'll get enough hints on how to use DIF within VisiCalc to pay for *The DIF File*.

If you use any of the programs Beil covers and think it'd be helpful to trade data between them, this book shows you exactly how to do so.

Finally, if you want to keep up with one of the more important areas of microcomputing—the transfer of data between software packages—you'll find this book more than worthwhile.

Gregory Glau
Prescott, AZ

Programming with Graphics

Garry Marshall
Spectrum, 1983
Prentice-Hall
Englewood Cliffs, NJ 07632
Softcover, 120 pp., $12.95

Programming with Graphics will get anyone into graphics on any of the commonly available microcomputers. The text is clearly written, straightforward and avoids all arithmetic beyond adding, subtracting, multiplying and dividing. However, understanding sine and cosine instructions will help enormously at some points.

Author Garry Marshall describes briefly how microcomputers handle graphics, and includes a few gripes about lack of standardization. Then he gets directly into the three types of graphics available: block, pixel and line.

Marshall covers color, high-resolution, printing graphics output, animation and motion and briefly mentions digitizers and plotters.

Diagrams, photographs and Basic programs are used in generous amounts to get ideas across. The extensive use of Basic code is a real asset that most readers will find convenient.

The last section deals with more advanced topics, such as the Apple Pascal turtle graphics and three-dimensional representations. The last chapter in this section ends with a statement of the requirements that may be sought in microcomputer graphics systems.

An appendix summarizes the graphics facilities provided by several microcomputers, plus some English brands not well-known in the United States.

The book does have its limits. Evidently, Marshall mainly has eight-bit microcomputers available for his work. Commodore is rather prominent in *Programming with Graphics*, with good discussions on the PET graphics and comparisons with graphics of other machines.

Marshall doesn't touch at all on 16-bit machines such as the IBM PC and its clones, which means that the emerging field of special graphics capabilities in these machines isn't covered.

Jim Derry
Akron, OH

Circle 90 on Reader Service card

Circle 128 on Reader Service card.

Circle 216 on Reader Service card.

Circle 252 on Reader Service card.

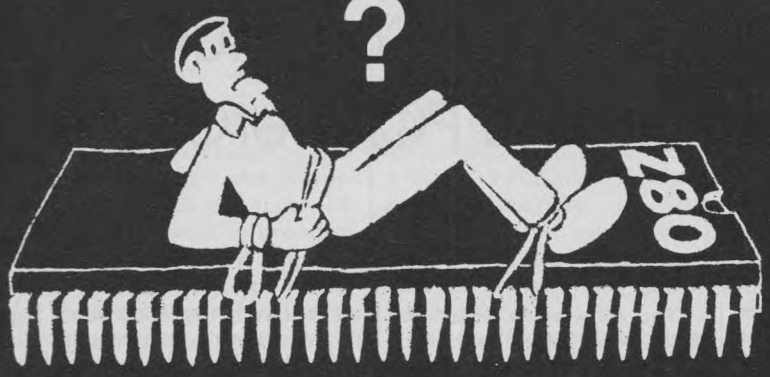
Circle 134 on Reader Service card.

Circle 37 on Reader Service card.

Circle 204 on Reader Service card.

TIED TO 8-BIT PROCESSING ?

Circle 295 on Reader Service card



ADD CO-POWER-88 AND BECOME IBM-PC COMPATIBLE

CO-POWER-88 is an 8088 coprocessor for Z80, CP/M computers. Its fast, 5.33 MHz, and has 256k RAM.

MSDOS comes with CO-POWER-88, and CP/M-86 is available as an option. Simple commands move system control from CP/M to MSDOS or CP/M-86 and back.

Most MSDOS, IBM-PC software runs on a CO-POWER-88 system. Run programs from an IBM-PC disk or transfer them to your system's MSDOS disk.

CO-POWER-88 doesn't replace your Z80, so you can still run CP/M... with an added feature, RAM disk. CO-POWER-88's RAM can be used as a high-speed simulated disk drive during CP/M processing.

Best of all, CO-POWER-88's affordable:

256k CO-POWER-88 w/MSDOS and RAM drive software $500.00

Available for: Actrix, ATR8000, Bigboard, Kaypro, Xerox and Zorba computers. And more to come.

Trademarks: ATR8000 and CO-POWER-88, SWP, Inc. IBM-PC, IBM Corp. Z80, Zilog. MSDOS, Microsoft Inc. CP/M and CP/M-86, Digital Research, Inc.

SWP Inc
MICROCOMPUTER PRODUCTS

2500 E. Randol Mill Road — 125
Arlington, Texas 76011
(817) 861-0421 (metro) 469-1181

Prices and specifications subject to change without notice. Specifications may vary for particular CO-POWER-88 systems.

Yes. I'd like to know more about CO-POWER-88.

Send to: SWP Microcomputer Products, Inc.
2500 E. Randol Mill Rd., #125
Arlington, TX 76011

#301B

name _____
street _____
city _____ state _____ zip _____

DEALER INQUIRIES WELCOME.

Ovation: Swift Multi-Application Tasks Enhanced Terminal Communications on IBM Zenith Z100 Utilizes PC/InterComm

Multi-User DBMS

MAG/base 4.0 is a database management software package for micros. It lets you simultaneously read and update a database without conflict or performance degradation.

With release 4.0, a single record is locked in during update and then released back into the database in its new form. While the record is frozen, it can still be read in its original form by other operators.

A B-tree scheme permits access to a node without locking the branches that extend beneath it. Extended relational capabilities require file-to-file manipulations.

Data can be manipulated logically and arithmetically from up to five files simultaneously. You can create new target files, update existing files and write reports and special forms.

The multiple-user features of release 4.0 extend to the other three members of MAG's database management family.

MAG/base is compatible with MP/M-86 and concurrent CP/M-86 operating systems as well as PC DOS, MS DOS, CP/M and CP/M-86. It can be used on Seiko, Altos, Molecular, Dynabyte, Televideo and other multi-user systems.

The software performs with either floppy or hard disk systems. Each package is sold with MAG's starter kit, which includes a primer with sample applications, a desktop reference for quick look-ups and an installation guide tailored to your operating environment.

The software is priced at $125 from MAG Software, Inc., 21054 Sherman Way, Suite 305, Canoga Park, CA 91303. Reader Service number 465.

IBM and Productivity

Ovation software is a program that is designed to let the IBM PC and PC-compatibles perform many productivity functions in just a few steps.

A common command environment lets you perform multiple business-related tasks without switching application modes. Thirty commands let you create spreadsheets, graphs, information files and documents as well as communicate with other computers.

The commands are summarized on a card designed to assist you with prompts, online help and tutorials. The commands let you print, send or receive information in any form without leaving the common command environment.

Ovation also provides immediate usability and customization capabilities with powerful macros that let you create multiple command sequences in order to perform repetitive tasks.

The program operates on the IBM PC and compatible 8086 or 8088 microcomputers running MS DOS. Ovation requires 256Kb of memory.

The program will run using two floppies, but a hard disk version on the IBM XT is preferable. Ovation will display color graphics using a standard IBM color card and color terminal.

Ovation will print to most industry print devices. A special feature lets you configure it to any printer as well. The program costs $795. Ovation Technologies, 770 Dedham St., Canton, MA 02021. Reader Service number 460.

MAG/base release 4.0 lets multiple users read and update a database without conflict.

PC/InterComm On the Z100

PC/InterComm, a data communications program, now operates on the Z100 personal computer.

PC/InterComm lets the Z100 emulate Digital Equipment Corp.'s VT100. Z100 operators can talk to DEC host minicomputers and run full-screen applications programs.

In addition, the software allows you to exchange data on-line with any network or computer. The system communicates at speeds up to 9600 bps and saves different setups for host computers by name.

PC/InterComm also lets you talk with other nonDEC mainframe computers as well as with micros. The package also provides other capabilities, such as the ability to "snapshot" the screen to a file or the printer at any time, and copy

incoming on-screen data to the printer. It also has the ability to error-check file transfers between personal computers running the system.

PC/InterComm has 30 programmable function keys, requires 128Kb of memory, one disk drive and an RS-232C ASCII communications port. It costs $99. Mark of the Unicorn, Inc., 222 Third St., Cambridge, MA 02142. Reader Service number 461.

Communications for Asynchronous Computers

Omniterm 2 is a terminal communications package that provides a full range of features designed to guarantee compatibility with any asynchronous computer, or modem.

Like its predecessor, Omniterm, the program operates through a command mode menu in which all communications parameters are grouped in logical categories. A scroll-back feature, easily accessible by using the PgUp and PgDn keys, allows you to view text that has been received in a similar manner while remaining connected to the remote system.

The program also includes complete translation tables, VT100 emulation, a status line and other features.

For file transfer, Omniterm 2 supports the checksum protocol, XModem. For character oriented transfer, Omniterm 2 supports several handshaking protocols and the ability to define pauses using XON/XOFF as a default.

In addition to using colors and sounds for instant feedback, Omniterm 2 comes with complete documentation, a keyboard overlay, an on-line help function and free telephone support. It retails for $245 from Lindbergh Systems, 49 Beechmont St., Worcester, MA 01609. Reader Service number 463.

DBMS for Unix

Personal Informix is a database management product for novice operators of Unix-based microcomputers.

The software is designed to be an interactive management environment that allows you to build databases easily.

Its clearly written screen prompts and menus allow you to build a database; create and use a transaction screen to add, delete and update records; and generate simple reports. You can also maintain and store data, such as mailing lists, telephone logs, calendars and address books.

The Personal Informix software is equipped with instructions for each command. After a command is executed, new information is available in the form of help screens that describe where you are in the program, the courses of action available and the results of each choice.

Personal Informix is priced at $495 for most Unix-based 16-bit microcomputers. It's manufactured by Relational Database Systems, Inc., 2471 E. Bayshore Road, Suite 600, Palo Alto, CA 94303. Reader Service number 466.

ClusterNet 3270 Enhances PC Express Communications

ClusterNet 3270 is an enhancement to the PC Express communications package from Intelligent Technologies International Corp.

The integrated hardware/software package permits an IBM PC to serve as an IBM 3274 cluster controller, which links as many as 12 IBM PCs with a mainframe. The new SNA software also provides "multisession" capabilities that enable a PC linked directly with a mainframe to display a variety of IBM 3278/9 terminal screens.

The enhancement allows an exchange of data between a cluster of PCs and a local or remote mainframe. It also provides each PC user in the cluster with concurrent access to several IBM mainframe programs.

The multisession capability lets you switch back and forth between mainframe sessions.

In addition to its new capabilities, the PC Express with SNA provides asynchronous protocols for DEC VT100/52 terminal emulation, as well as Intelligent Technologies' standard telephone management software. The package's automated telephone capabilities include autoanswering, autodialing of numbers drawn from a database telephone directory, a full-screen text editor, remote terminal access, and electronic mail.

ClusterNet is now available as a standard feature of the PC Express package, which is priced at $1295 from Intelligent Technologies International Corp., 151 University Ave., Palo Alto, CA 94301. Reader Service number 464.

Personal Informix is an interactive database management product designed for Unix-based 16-bit microcomputers.

Apple Integrated Software

This new integrated software package for Apple II and Apple III computers combines word processing, database management and financial modeling.

Called AppleWorks for the Apple II and III E-Z Pieces for the Apple III, the software can be used in a business or home environment. Each function operates with the same speed as many stand-alone applications. The program provides three levels of integration designed to further increase efficiency.

The word processing application displays documents on the screen as they will appear when printed, including centered, indented text and page breaks.

The spreadsheet program provides a large work area of 999 rows by 127 columns for financial models.

The database manager has up to 30 categories of records in alphabetic, numeric, date or time order. The program is memory-based, so sorting should be speedy.

AppleWorks and III E-Z Pieces include a menu-driven desktop manager that handles utility functions such as loading and saving files, formatting disks and specifying printer information.

AppleWorks costs $250 and requires an Apple IIe PC with 64Kb of memory and an 80-column card, one floppy disk drive and a monitor. Apple's Extended 80-Column Card for the IIe, which provides an additional 64Kb of memory, and a printer are recommended.

The requirements for III E-Z Pieces are an Apple III computer system with 256Kb of internal memory. It is priced at $295.

Both products are manufactured by Apple Computer, Inc., 10260 Bandley Drive, Cupertino, CA 95014. Reader Service number 468.

Microsoft Windows

Microsoft Windows is a new software environment for developing and running applica-

tion programs that use bit-mapped displays and mouse painting devices.

Microsoft Windows is designed to free elaborate bit-mapped application programs from device dependence. The software programs are portable to any system running the Windows.

The Microsoft Windows environment supplies user interface functions that are common to graphics-based application programs. Under Microsoft Windows, each application program is viewed through its own window and multiple programs may be viewed simultaneously.

Libraries of user interface and graphics functions, including the ability to exchange data, are shared by all clients.

Microsoft Windows has two parts: the window manager and the graphics device interface (GDI). The window manager draws the window for each client and manages the screen as a whole, using an automatic window layout technique.

The window manager is event-driven; it passes hardware-level events to its clients and the clients supply response procedures. It also provides library-to-user interface functions.

The library includes menus, property sheets, scroll bars, universal window commands, error handling, data interchange and automatic window layout.

To display its graphics-based interface features on a screen, the window manager calls the graphics device interface. The GDI can also be called directly to generate graphics.

The hardware requirements for Microsoft Windows are 192Kb of RAM, a mouse, two floppy disk drives and a bit-mapped display. A hard disk is not required.

Microsoft Corp., 10700 Northup Way, Bellevue, WA 98004. Reader Service number 472.

A Powerful Spreadsheet

PractiCalc II is an advanced, spreadsheet program for the Apple II and IIe microcomputers.

PractiCalc II has all the features of its original program, PractiCalc. Its 15 new features include advanced editing capabilities, variable column width, upper- and lowercase entry and printing for the IIe. It has the ability to do long labels and automatic and manual recalculation and is equipped with an on-screen default menu.

In addition to traditional spreadsheet functions, PractiCalc II performs full alpha and numeric sorting and numeric searching and prompts for entry during calculations and printing of list formulas.

PractiCalc II costs $69 from Micro Software International, The Silk Mill, 44 Oak St., Newton Upper Falls, MA 02164. Reader Service number 471.

IBM Merges Four in One

Jack2 is a second-generation software package for the IBM PC that merges four major functions into one system.

It lets you word process, manage a database, create spreadsheets and design graphs.

All of Jack2's functions can be used in a single document and are available in one mode, on one screen. Jack2 supports a variety of text handling capabilities, including nine user-definable text highlighting options.

The software creates documents up to 255 characters wide and allows independent editing within user-defined columns. Documents on separate disks can be linked for consecutive printing. Documents containing one or more fields can be turned into a series of records.

Jack2 allows column and row fields to be placed within your text or presented as a spreadsheet. Up to 1000 columns and rows can be used within field definitions to make a field calculate and/or display a value from another field. A host of built-in mathematical operations and functions are provided.

Graphs can also be created within a text document and can be designed to chart values from a spreadsheet or from other fields within a document. They are automatically updated when new values are entered into the field being graphed.

A tutorial disk and reference manual are provided with the package. A Jack2 data disk is included, containing data for use with the tutorial lessons. This disk also contains templates that can be used and modified for record keeping purposes.

Jack2 is priced at $495 from Business Solutions, Inc., 60 East Main St., Kings Park, NY 11754. Reader Service number 469.

Apple's College Explorer

College Explorer for the Apple II Plus and IIe helps you choose a college based on your interests and skills. The program features 2700 colleges and universities throughout the United States.

The program's file, which will be revised each year, is based on College Board data collected annually for publication in *The College Handbook*, the Board's guide to American colleges and universities.

Students search the file using more than 500 options for 12 important college features such as curriculum, geographic location and enrollment size. By specifying any combination of the features listed, students can create a personal college profile.

College Explorer costs $149 and comes with four floppy disks, a counselor's manual, two student manuals, a student worksheet for counselor duplication, a technical specifications card and a copy of *The College Handbook*, 1983-84. It is manufactured by College Board Publications, Box 886, Department A67, New York, NY 10101. Reader Service number 473.

Amperware for Applesoft Basic

Amperware enhances the capabilities of the Applesoft Basic programming language.

Seventeen new Basic commands improve the performance of keyboard input, disk I/O, searching, sorting and formatted printing.

The &Input statement makes video form generation simple. It allows entry of upper- and lowercase characters, commas, colons, quotation marks and special characters. It also controls the type and length of each user response and provides full in-line editing capabilities.

Disk I/O can be increased by a factor of 20. The screen commands make it possible for the same Basic code to work with any hardware combination of Apple II or IIe with a 40- or 80-column board.

Amperware includes a detailed manual that contains a discussion of basic and advanced programming techniques, plus a complete reference section and index. The software comes on disk in standard DOS 3.3 format and is not copy protected.

Amperware is priced at $49.95 plus $2 shipping from Scientific Software Products, Inc., 5726 Professional Circle, Suite 105, Indianapolis, IN 46241. Reader Service number 475.

Commodore's SuperTerm

SuperTerm is an intelligent terminal program that allows Commodore-64 owners to upload and download with a wide range of business and university mainframe computers as well as hobby bulletin boards and CompuServe.

SuperTerm combines a unique combination of often-requested telecommunications features, including:

- A text editor that can manipulate up to 18.4Kb of information at once (on- or off-line) with insert, delete, erase and merge features.
- The ability to display text in 40, 80 or 132 columns using a fast side-scrolling technique.
- Continuous on-line printing for owners of parallel printers who have Midwest Micro's Smart ASCII Plus interface, as well as off-line printing for users with Midwest Micro's Smart ASCII Plus interface and off-line printing using other interfaces or printers, such as the VIC-1515 or VIC-1525.
- Quick saves of large amounts of incoming information and programs to disk to reduce long-distance tele-

Get The Whole Picture

Discover MICROCOMPUTING®

Now you can develop your knowledge of computing instantly. Microcomputing uncovers important trends and new discoveries...then shows how they'll affect you.

Other multi-system magazines are loaded with filler stories. Not Microcomputing. Every issue is alive with useful programs, money-saving projects, instructive columns, and exciting articles. Here are just a few:

- Overview examines industry trends and new technologies in-depth.
- new-product announcements explore state-of-the-art products before they reach the stores.
- extensive buyer's guides save you money as you comparison-shop from home.
- candid reviews of software, hardware, and books ensure that your next purchase will be a sound investment.
- hardware projects help you modify your system to fit your changing needs.
- software conversions widen the scope of your machine.

With all this at your fingertips, you could save the cost of your subscription with one wise purchase.

Subscribe to Microcomputing today. A full year is only $24.97. Just fill out the coupon below or the attached order card and return it right now to: Microcomputing Subscription Department, PO Box 997, Farmingdale NY 11737. For even faster service, call toll free:

1 (800) 258-5473 (In New Hampshire call 1-924-9471)

You risk nothing. If you don't like your first issue for any reason, simply write "cancel" across the invoice and return it. You won't owe a thing.

Subscribe today...and get a 13th issue FREE when you include payment or charge it on your Mastercard, Visa, or American Express.

YES! Give me the whole picture. Send me 12 issues of Microcomputing for $24.97. I understand that with payment enclosed or credit card order I will receive a FREE issue, making a total of 13 issues for $24.97.

☐ Check enclosed ☐ MC ☐ VISA ☐ AE ☐ Bill me

Card #_____ Exp. date_____

Signature_____

Name _____

Address _____

City_____ State_____ Zip_____

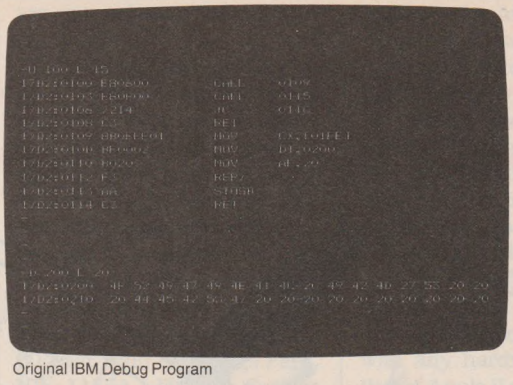
MICROCOMPUTING®

343F7

PO Box 997, Farmingdale NY 11737

Canada & Mexico $27.97, 1 yr. only, US funds drawn on US bank.
Foreign surface $44.97, 1 yr. only, US funds drawn on US bank.
Please allow 6-8 weeks for delivery. Foreign Airmail please inquire.

The difference



Original IBM Debug Program

is Mylstar's Symbolic Debugging Program*

```
*
*
*U \GETLIN L 13
2010:0100 E80600        GETLIN: CALL    CLRBUF
2010:0103 E80F00                CALL    GETCHR
2010:0106 7214                  JC      ERROR
2010:0108 C3                    RET
2010:0109 8B0EFE01      CLRBUF: MOV     CX,[BUFLEN]
2010:010D BF0002                MOV     DI,Offset BUFFER
2010:0110 B020                  MOV     AL,20
2010:0112 F3                    REPZ
2010:0113 AA                    STOSB
2010:0114 C3                    RET
*
*
*
*
*D \BUFFER L [\BUFLEN]
2010:0200  4D 59 4C 53 54 41 52 20-53 59 4D 42 4F 4C 49 43
2010:0210  20 44 45 42 55 47 47 45-52 20 20 56 20 31 2E 31
*
*
```

Mylstar Symbolic Debugger V1.1

The plain and simple difference is that Mylstar's Symbolic Debugging Program speaks to your IBM PC in a language you both can understand, plain and simple.

Employing the same command structure, it allows you to use symbol names, mathematical expressions, batch files, on-line help, multi-command macros and other time-saving entries.

It's the enhancement to the *IBM Debug Program* you've been looking for—because it fills in the gaps—shortening the frustrating debugging process by as much as 50%—leaving you more time to do the work you need to do and the work you want to do, plain and simple.

Mylstar's Symbolic Debugging Program has been programmer-tested for over a year at Mylstar Electronics, Inc., (formerly D. Gottlieb & Co.), designers of the video arcade game, Q*BERT™.

TO ORDER...
Call (312) 562-7400 or mail coupon today.

**Designed for IBM PC-DOS 1.1 with 128K RAM minimum*

MYLSTAR ELECTRONICS INC.

Circle 312 on Reader Service card.

165 West Lake Street
Northlake, Illinois 60164

A Columbia Pictures Industries Company

Mylstar Electronics, Inc.,
165 W. Lake St., Northlake, IL 60164

Please send me Mylstar's Symbolic Debugging Program for use with the IBM PC computer. Enclosed is $125, plain and simple.

☐ Check ☐ Money Order 117

NAME _____

FIRM _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

Illinois residents add 7% sales tax
Allow 2-4 weeks for delivery

phone charges and connect fees.

- Autodial and autoanswer using automodems from Commodore, HES or Microperipheral Corp.

SuperTerm has the ability to mimic other terminals, such as the VT100. The program contains 52 user-defined functions and a wide range of settings for baud rates, parity, word size, stop bits and other communications parameters.

SuperTerm comes with a hardware module that plugs into the cartridge port, plus extensive documentation. SuperTerm on disk retails for $149.95 and Smart ASCII Plus for $59.95. Both products are manufactured by Midwest Micro, Inc., 311 W. 72nd St., Kansas City, MO 64114. Reader Service number 477.

Compose and Play Music on Your Commodore-64

MusicCalc I turns the Commodore-64 into a professional three-voice synthesizer with fully interactive real-time sequencing, slide controls, modulators and transposers.

MusicCalc I lets you play along with preprogrammed melodies or create and store your own melodies for later playback. You can compose and perform in styles ranging from classical to new wave to Japanese to Latin to African.

MusicCalc I eliminates the usual prerequisite training required to learn how to play and compose music. It also functions on a level suitable for professional recording artists.

The program is priced at $74.95 from Waveform, Inc., 1912 Bonita Way, Berkeley, CA 94704. Reader Service number 478.

Aerobics with Atari And Commodore-64

Aerobics software lets Atari and Commodore-64 owners customize their fitness programs.

Selecting from 18 preset exercise segments that vary from 30 to 90 minutes, you can work on overall conditioning or concentrate on specific body parts. Exercises are graded for beginner, intermediate and advanced levels. With two available speeds, you can set your own pace.

An animated aerobics instructor and captioned instructions guide you in your movements and offer such helpful reminders as "breathe!". Each segment has four parts: warm-up; aerobics; body parts conditioning; and cool-down. Nine different tracks of up-beat computer-synthesized music provide a background for your fitness program.

Aerobics retails for $44.95 and is manufactured by Spinnaker Software, 215 First St., Cambridge, MA 02142. Reader Service number 479.

ProKey Gets Customized

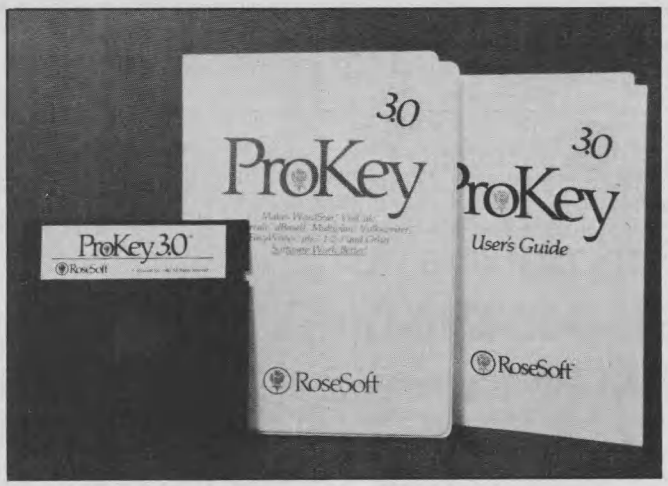
ProKey 3.0 is an enhanced version of ProKey that can be used to customize both off-the-shelf software and self-written programs.

After you assign a definition to a key, ProKey is ready to run. Key definitions can contain just a single character or more than 12,000 characters of text or commands.

One of the new features of ProKey 3.0 is a menu-driven program that makes it easy to change key locations. You can create a customized keyboard layout or use the alternate layout files provided on every ProKey disk.

A file is included for the Dvorak keyboard layout. Another file mimics the layout of the IBM Selectric keyboard. By reversing the locations of keys, ProKey answers the complaints of many touch-typists who have been struggling with the layout of the IBM PC keyboard.

Correction: The Totl.Text software package *(Microcomputing,* December, 1983) should have been identified as a product of TOTL Software, Inc.



ProKey lets you create macros to customize both off-the-shelf software and self-written programs.

With ProKey 3.0, a macro can be assigned to any key, either alone or in combination with the Shift, Alt or Ctrl keys, for a total of over 250 available keys.

ProKey 3.0 is compatible with most IBM PC software. The ProKey disk includes files that streamline WordStar, Lotus 1-2-3, Visicalc and dBaseII.

ProKey 3.0 also offers an online demonstration and a rewritten user's guide. The guide contains step-by-step instructions and a new index and glossary.

ProKey 3.0 is priced at $129.95 from RoseSoft, 4710 University Way N.E., Suite 601, Seattle, WA 98105. Reader Service number 474.

Edited by Sheila Wright

On the Portable Battlefield:

Panasonic's PC: Senior Partner

Gavilan's 16-Line Display Portable

STM Portable PC: IBM Compatibility

IBM-Compatible Commuter

The Commuter is claimed to be the industry's first 16-pound, 16-bit IBM-compatible personal computer.

It features 128Kb of memory (expandable to 512Kb), MS DOS 2.1, an 83-key keyboard with all IBM functions and an identical layout. It also features a 5¼-inch, double-sided, double-density 360Kb floppy disk drive (an additional drive is optional).

The commuter also boasts bit-mapped color graphics capability, a parallel printer port, an RS-232C serial port, RGB direct-drive output for high resolution monochrome or color monitors, composite video output, power-up diagnostics and a connector for the IBM expansion chassis that lets you use IBM-type memory cards and peripherals. An 80-character by 16-line LCD is optional.

The Commuter is priced at $1995 from Visual Computer, Inc., 135 Maple St., Marlboro, MA 01752. Reader Service number 483.

Panasonic's Portable

Senior Partner, model RL-H7000, is an IBM PC-compatible portable from Panasonic.

The system runs under MS DOS 2.0 and runs IBM-compatible software. It features a built-in thermal printer and a double-sided, double-density, 320Kb disk drive; a second 320Kb disk drive is optional. Senior Partner contains 128Kb of memory (expandable to 512Kb) and a nine-inch screen.

The computer uses an 8088 microprocessor; a socket is provided for an optional 8087 math coprocessor.

The keyboard accepts IBM overlays. The RGB monitor output provides built-in color graphics. A Centronics parallel port is provided as well as an RS-232C serial port. An optional expansion slot lets the machine use IBM-type boards.

Software bundled with Senior Partner includes WordStar, VisiCalc, pfs:File, pfs:Graph, pfs:Report and GW Basic. The system retails for $2495 from Panasonic, One Panasonic Way, Secaucus, NJ 07094. Reader Service number 480.

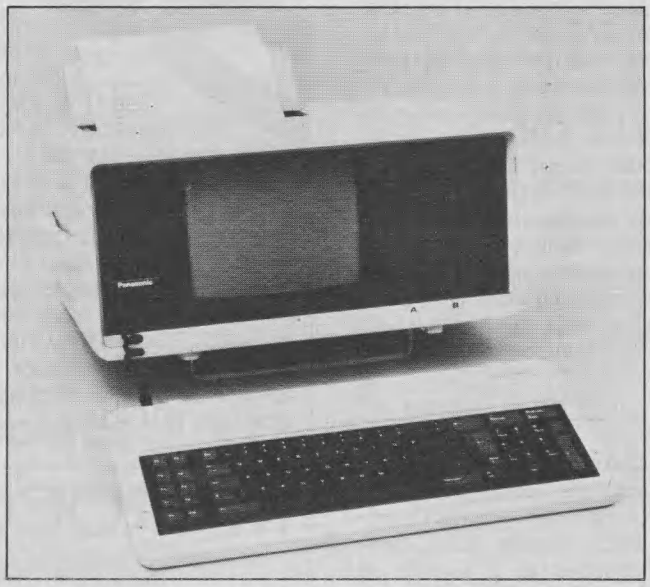
Kalglo Electronic's Standby Power System

Line-Saver, Model LS-240, is a standby uninterruptible power system from Kalglo Electronics Co., Inc.

The system is designed to provide standby power in the event of a power failure and is available in 120/240 volt, 60/50 Hz, with 240 VA, 150 watts capacity. The unit utilizes pulse width modulation technology to regulate the ac output voltage for greater efficiency under various load conditions.

Line-Saver provides five to ten minutes of power at full load, 20 to 25 minutes at half load and 35 to 40 minutes at one-third load. It is furnished with an internal 12V sealed rechargeable battery, four Spike-Spiker voltage-surge-protected and EMI/RFI-filtered ac outlets, audible and visual power failure warning system, test mode indicator and switch, replaceable external fuses, and external 12VDC battery connectors to allow for mobility and extended-time operation.

The Line-Saver comes with a detachable six-foot, three-prong, heavy-duty grounded cord set with a CEE-22 connector. It retails for $395 from Kalglo Electronics Co., Inc., Dept. LS, 6584 Ruch Rd., Bethlehem, PA 18017. Reader Service number 489.



Panasonic's IBM-compatible portable features a built-in thermal printer.

MicroStandard's Portable PC

The M3000 is a transportable computer from MicroStandard Technologies, Inc.

The system runs under four possible operating systems. The base unit is supplied with CP/M Plus installed; MS DOS, CP/M-86 and Unix are optional and require optional processor cards.

Three optional expansion card cages are available: STD bus with four open slots, VME bus with two open slots and S-100 bus with two open slots.

Three possible disk drive configurations provide a choice of storage capacities. The basic model has one 376Kb double-sided, double-density 5¼-inch drive. Options include two 376Kb 5¼-inch floppies or a combination of a 10Mb hard disk with one 376Kb floppy.

The detachable IBM-style keyboard has 93 keys, including 14 programmable function keys and an 18-key calculator pad. The calculator pad supports a built-in, full-function calculator that displays on the computer's nine-inch screen.

The basic M3000 retails for $1645. The two-drive model sells for $1995 and the hard disk-equipped model, $2895. The system is manufactured by MicroStandard Technologies, Inc., Box 319, New Lebanon, OH 45345. Reader Service number 485.

STM Portable: Another IBM Compatible

The STM Personal Computer is a 17-pound IBM PC-compatible portable computer that features a 16-line LCD, a 40-column thermal printer with graphics capabilities, an internal modem, a built-in speaker phone and software.

The STM PC provides an expanded display capability for standard video monitors. It supports windowing software and allows you to choose between a 25-line by 80-column or 25-line by 132-column display with 640 by 400 resolution.

The STM PC uses an 80186 microprocessor and runs MS DOS 2.0. It has 256Kb of memory, which is expandable to 512Kb, and is equipped with two double-sided, double-density, one-megabyte floppy disk drives. The computer's direct-connect modem is programmable between 300 and 1200 bps and features auto dial/autoanswer capabilities.

In addition to the standard parallel printer port, the unit has two programmable serial ports, a composite video display, a SASI-type hard disk interface and an edge connector that brings out the I/O signals for an external expansion box.

The STM PC also has a detachable keyboard with 94 keys, including a numeric keypad and ten function keys. The machine comes with integrated word processing, spreadsheet, database and graphics software.

The STM PC retails for $3000 from STM Electronics, 530 Middlefield Road, Suite 250, Menlo Park, CA 94025. Reader Service number 482.

A Bigger Display For Gavilan

The Gavilan portable micro now features a 16-line liquid crystal display, replacing the 8-line display that the machine was introduced with.

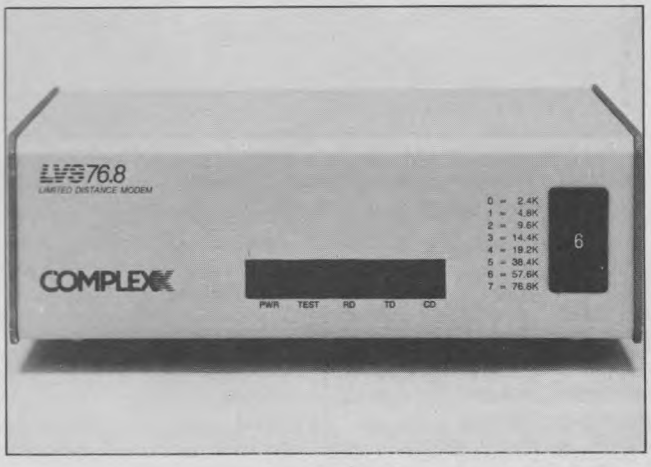
Gavilan has also introduced a second version of the machine. Called the Gavilan SC, the new computer is similar to the original Gavilan and includes 64Kb CMOS nonvolatile memory and a built-in 3½-inch microfloppy. It lacks the Gavilan's internal modem, MS Basic capability and the Gavilan integrated application environment.

Both machines feature an integrated touch panel for cursor control, an RS-232C serial port and video monitor output interface. Both run MS DOS.

The Gavilan retails for $3995; the Gavilan SC, $2995 from Gavilan Computer Corp., 240 Hacienda Ave., Cambell, CA 95008. Reader Service number 481.

The STM Personal Computer uses Intel's 16-bit 80186 processor and runs under MS DOS 2.0.

The Gavilan's integrated touch panel lets you control the cursor for menu item selection and on-screen data manipulation.



The LVS 76.8 from Complexx Systems is a limited-distance modem that can operate at up to 76,800 bits per second.

Interact or Stand Alone with Micro Plus

Micro Plus is a CP/M-based desktop computer from Computer Designed Systems, Inc.

The system can communicate with the company's Adviser series of interactive computers or it can operate as a stand-alone CP/M-based system.

Micro Plus is available in either a monochrome or color version. The system comes with an 80-column printer, a 12-inch screen and one 5¼-inch disk drive. Its storage capacity ranges from 500Kb to 2Mb, depending on configuration; optional hard disk storage is available. Micro Plus has a 96-key typewriter keyboard that includes a 10-key numeric keypad.

Micro Plus retails for $2000 from Computer Designed Systems, Inc., 10911 Olson Memorial Highway, Minneapolis, MN 55441. Reader Service number 488.

High Performance Modem from Complexx Systems

The LVS 76.8 is a limited-distance modem that uses a new modulation technique to allow transmission speeds of from 2400 bps to 76,800 bps.

The LVS lets you select from eight synchronous speeds using a thumbwheel switch on the unit's front cover. At its 76.8Kbps speed, LVS 76.8 can send data 16,250 feet using 22-gauge wire. Greater distances can be attained by transmitting at slower speeds or by using heavier-gauge wire.

The LVS is programmed to provide standard EIA and line loopback testing. It has LED indicators for power, data transmit, data receive, carrier detect and testing. It's available with either an RS-232C serial interface for $650 or with a V.35 interface for $725. Complexx Systems, Inc., 4930 Research Drive, Huntsville, AL 35805. Reader Service number 506.

Expandable Execuport XL Series

The Execuport XL computers are two expandable portables from Computer Transceiver Systems. Both machines have a nine-inch green phosphor screen that can display 132 columns by 24 lines, and two double-sided, double-density, 5¼-inch disk drives with 800Kb of formatted storage each. Both keyboards have 22 user-programmable function keys.

The Execuport XL contains a Z80, 80Kb of memory and runs CP/M. The Execuport XL+ adds a 16-bit 80186 processor, 128Kb of memory, MS DOS and the Perfect series of productivity software. The XL can be field-upgraded to the XL+.

Future options will include a 300/1200 bps intelligent modem, a telephone handset, a 200 cps printer and a hard disk.

The Execuport XL costs $2695; the XL+ goes for $3495. Both are available from Computer Transceiver Systems, Inc., PO Box 15, East 66 Midland Ave., Paramus, NJ 07652. Reader Service number 484.

Integrate PC 1000

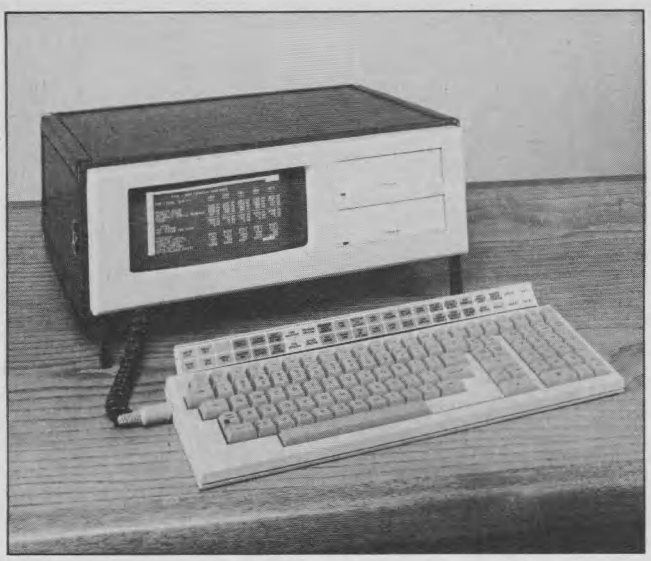
The PC 1000 Personal Instrument System is a laboratory-grade intelligent test and measurement equipment system for the IBM PC.

The heart of the system is the PC 1810 expansion chassis, which extends the computer's bus and provides isolated power supply voltages to avoid ground-induced errors that often occur when an instrument is plugged directly into the computer chassis. A special disk operating system lets you access and control each test module using the keyboard or through Basic programs.

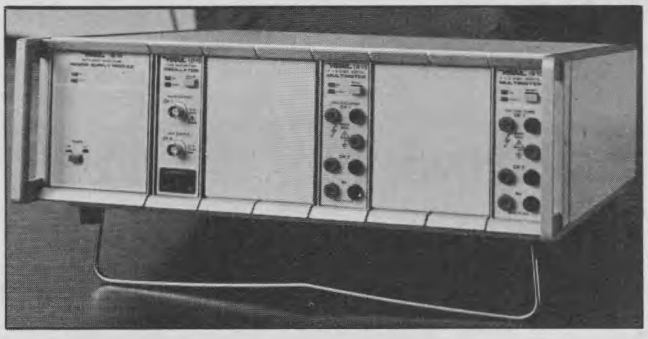
Test modules currently available include a digital multimeter ($790), a low-distortion oscillator ($990) and an IEEE-488 interface ($390). Additional modules, including a digital oscilloscope, FFT spectrum analyzer and programmable power supply, are due soon.

The system's PC 1810 expansion chassis retails for $1890. Seven PC 1000 series modules can be plugged into the 1810. This expansion chassis extends the IBM PC bus and provides the isolated power supply voltages needed to avoid ground induced errors.

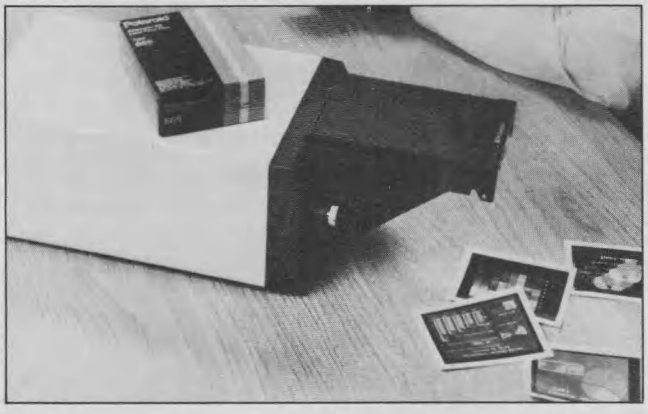
The PC 1000 Personal Instrument System is manufactured by Vistar Corp., 13740 McCormick Dr., Tampa, FL 33624. Reader Service number 491.



The Execuport XL, a Z80-based machine, can be upgraded to the Execuport XL+, an 80186-based micro that runs MS DOS.



The PC 1810 expansion chassis allows up to seven intelligent test equipment instrument modules to be plugged into it.



Polaroid's Palette computer image recorder can produce color slides and prints from even a monochrome screen.

Shoot Your Screen With Polaroid's Palette

Polaroid's Palette is an image-recording system for IBM PCs and Apple II Plus and IIe computers.

Featuring an exposure unit, 35mm autoadvance camera back, Polaroid 35mm autoprocess transparency system hardware, Polaroid 3¼ × 4¼-inch instant print camera back, connecting cables and software, Palette lets you produce high-quality 35mm slides and instant prints or transparencies of screen graphics.

Containing a flat-faced monochrome video screen with a red, green and blue filter wheel, Palette lets even monochrome display computers with graphics capability produce color hard copy. The computer and software match exposure parameters to the film being used while letting you control color selection and location.

Palette is priced at $1499 from Polaroid, 575 Technology Square, Cambridge, MA 02130. Reader Service number 511.

A Combination Hard Disk System For IBM PCs

Diskit Combo is a hard disk subsystem for the IBM PC that uses a 10Mb fixed disk and a 5Mb removable cartridge for fast disk-to-disk backups of the fixed disk.

Diskit Combo is also available with dual fixed and/or dual removable drives. It runs under MS DOS 2.0 and draws an average of 1.10 amps. Software is provided on a floppy disk and is compatible with current MS DOS backup and restore utilities. Higher-capacity fixed-only drives providing up to 150Mb are also

available, as are versions for other computers.

Diskit Combo is priced at $2795 with the 5Mb removable cartridge and at $1495 without it. It is manufactured by Systems Peripherals Consultants, 9747 Business Park Ave., San Diego, CA 92131. Reader Service number 503.

A 16-bit Upgrade For TRS-80s

MicroMerlin is a self-contained, 16-bit upgrade for TRS-80 models I, III, 4 and 4P, and LNW model 1 computers.

Two available operating systems, MS DOS 2.1 and CP/M-86, let your TRS-80 read and write IBM PC-format disks and execute many IBM PC programs. MicroMerlin contains an 8088 processor that runs at 5 MHz, 128Kb of memory (which is expandable to 768Kb), an RS-232C serial port, a parallel port and power supply. Available options include memory expansion, an 8087 math coprocessor and color graphics.

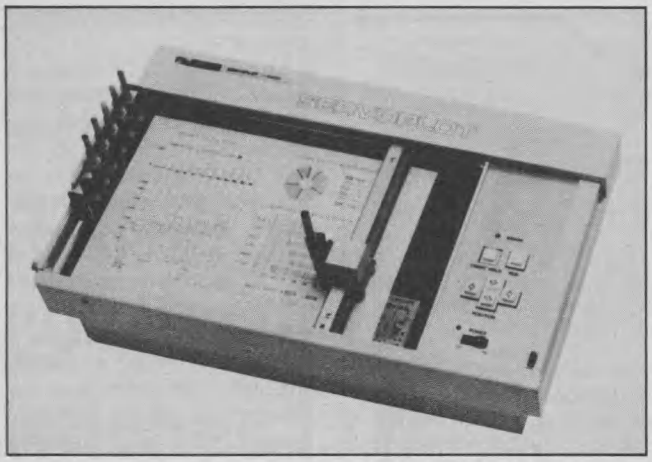
MicroMerlin can also be used as a printer spooler and as a disk emulator. It sells for $995 with one operating system and is manufactured by Micro Products Engineering, Inc., 10810 W. Washington Blvd., Culver City, CA 92030. Reader Service number 510.

A Sharp Dot Matrix Printer from Smith-Corona

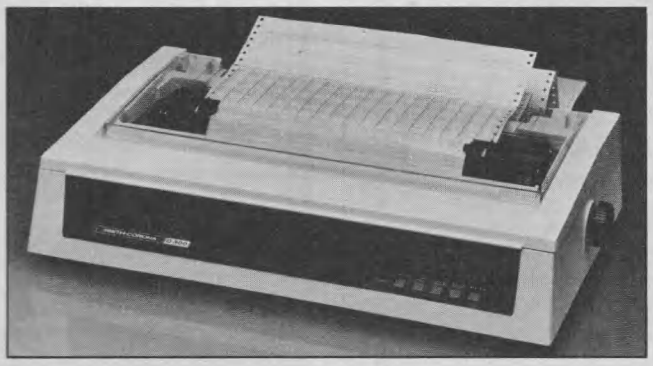
Smith-Corona's D-300 dot matrix printer features six different pitches, choice of emphasized or elongated print, proportional spacing, an italics character set, bit-image graphics, superscript/subscript capability, bidirectional printing, friction and tractor feed and a self-test mode.

The printhead boasts a character matrix of 9×8 (standard) to 17×16 (near letter-quality) and true descenders. The D-300 also features a 2Kb print buffer, horizontal resolutions of 60, 72 and 120 dots per inch, built-in parallel and serial interfaces and a 140 cps printing speed.

The D-300 retails for $795.



Western Graphtec's DA 8400 is a six-pen, intelligent x,y plotter that plots at up to 16 inches per second.



Smith-Corona's D-300 dot matrix printer ($795) features 140 cps printing speed and bit-image graphics capability.

Smith-Corona, 65 Locust Ave., New Canaan, CT 06840. Reader Service number 496.

Sony's Plug-Compatible Drives

Sony Data Products has introduced a 5¼-inch plug-compatible version of its 3½-inch microfloppy drives capable of replacing standard minifloppy drives without any interfacing problems.

The data-transfer rate (250K bits per second in double-density mode) and 34-pin connector used in the new microfloppy drives allows interchangeability with 5¼ drives. The drives are available in single-sided (OA-D33V) and double-sided (OA-D33W) versions and in double-density mode and provide 500Kb and 1Mb storage capacities, respectively.

The single-sided version sells for $250; the double-sided version, $300. Sony Communications Products Co., Sony Drive, Park Ridge, NJ 07656. Reader Service number 501.

Western Graphtec's Intelligent Plotter

The DA-8400 is a six-pen, intelligent x,y plotter that features a dc servo motor and RS-232C, 8-bit parallel or IEEE-4888 interfaces.

The DA-8400 offers a maximum printing speed of 16 inches per second and is available in both A3 size (11×17 inches) and A4 size (8½×11 inches). It's also available in both flatbed and rollfeed models. The flatbed version features an electrostatic paper hold-down; the rollfeed model secures the paper using back-tension rollers.

The DA 8400 uses either water- or oil-based fiber-tip pens. Pen selection is automatic. The A4 model sells for $1850; the A3 roll feed model, $2350. Western Graphtec, Inc., 12 Chrysler St., Irvine, CA 92714. Reader Service number 498.

Mountain Computer's Internal Hard Disk For the IBM PC

Mountain Computer, Inc. has developed an internal hard disk system for the IBM PC.

Comprising a 12.76Mb half-high hard disk (formatted to 10Mb), a controller card and software, the Mountain Computer upgrade kit is reportedly the only system capable of supporting a hard disk using the PC's existing power supply while emulating the IBM's XT disk controller operating under MS DOS 2.0 and all other XT operating systems, including CP/M-86 and UCSD Pascal.

The hard disk upgrade system sells for $2195 from Mountain Computer, Inc., 300 El Pueblo Road, Scotts Valley, CA 95066. Reader Service number 504.

A Low-Cost Intelligent Modem

Operator 103 is an intelligent autodial/autoanswer modem from TNW Corporation.

The system uses two LSI chips—a single-chip microprocessor or all control functions and an integrated modem chip that includes necessary filters. Operator 103 provides normal and fast-speed pulse dialing and runs at 300, 200, 150 and 110 bps. The modem comes with an instruction manual, power supply and modular telephone cord.

Operator 103 is warranteed for two years and retails for $189. TNW Corporation, 3444 Hancock St., San Diego, CA 92110. Reader Service number 505.

Seequa's Powerful Desktop Computer

The Seequa/XT is an IBM-compatible desktop computer from Seequa Computer, the

manufacturers of the Chameleon portable.

The Seequa/XT's dual-processor design (8088 and Z80) allows it to run the popular 16-bit software written for the IBM PC as well as the thousands of 8-bit CP/M-80 programs. Its 256Kb of memory is expandable to 640Kb and its 8Kb of ROM is expandable to 48Kb.

The Seequa/XT features a 10Mb hard disk, a 320Kb 5¼-inch disk drive and an 83-key keyboard. Color graphics are also included with color resolution of 320 × 200 pixels and black and white resolution of 640 × 200 pixels.

Other features include five expansion slots with four available, a parallel printer port and a serial communications port. The machine also comes with PerfectWriter, PerfectSpeller, PerfectCalc, MS DOS, GW Basic and the Condor I database management system.

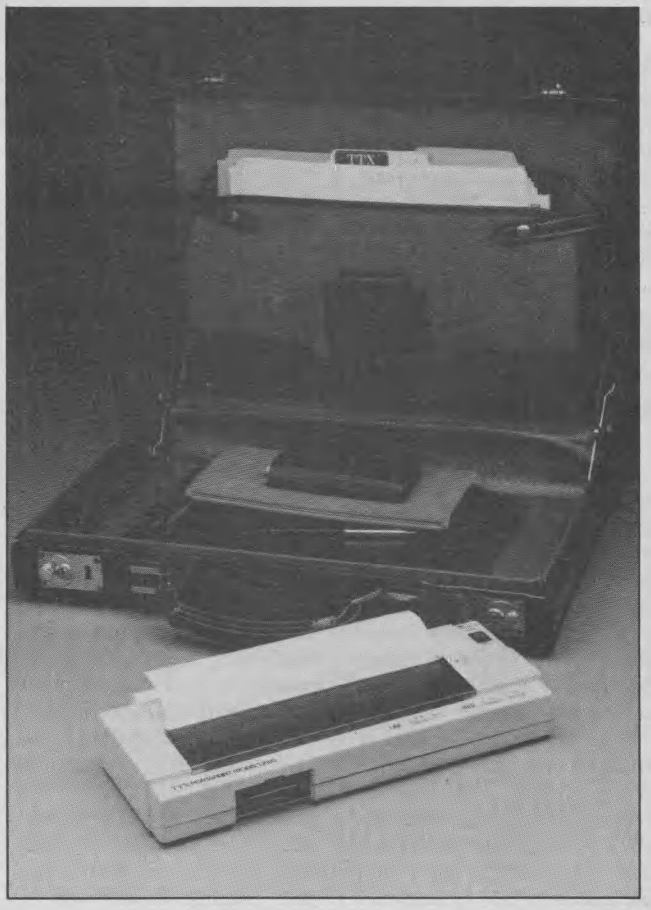
The Seequa/XT retails for $3995 from Seequa Computer Corp., 8305 Telegraph Road, Odenton, MD 21113. Reader Service number 486.

Print On Route

The TTX 1280 Portaprint is a three-pound, battery- and ac-powered, 80 to 132-column thermal printer from Teletex Communication Corp.

The printer uses rechargeable or replaceable 6V batteries that provide about 5000 lines of print. Portaprint prints 40 characters per second under battery power and 80 characters per second under ac power.

Using its 5×7 printhead, Portaprint can produce a number of different character sizes and densities, including bold, shadow, oversized and condensed. The printer also supports line and dot graphics. It retails for $199. Teletex Communication Corp., 3420 East Third Ave., Foster City, CA 94404. Reader Service number 493.



The TTX 1280 Portaprint is a three-pound, ac- or battery-powered thermal matrix printer for portable and handheld computers.

Circle 189 on Reader Service card.

EPSON, CITOH, OKIDATA, TABCO, TRS-80* Printer Switches

SAVE A BUNDLE

Order Toll Free 1-800-874-1551
FLA Residents 904-438-6507 collect

H&B SALES CO.

704 W Michigan Ave;
P.O. Box 8098
Pensacola, FLA 32505

*TRS-80 is a trademark of Tandy Corporation

Circle 319 on Reader Service card.

NOW THERE ARE THREE!

The Dayton HAMVENTION will present three awards to selected recipients at the 1984 HAMVENTION on April 27, 28, 29, 1984. In addition to the AMATEUR OF THE YEAR and the SPECIAL ACHIEVEMENT awards, a third award for TECHNICAL EXCELLENCE will be given annually for outstanding accomplishment specifically oriented to the technical aspect of amateur radio.

Nominations are requested for each of these prestigious awards. The deadline for submission is April 1, 1984. Write for additional information.

AWARDS COMMITTEE
1984 Dayton HAMVENTION
P.O. Box 44
Dayton, Ohio 45401

Circle 235 on Reader Service card.

Commodore 64 and VIC-20

SuperTerm

$149.95

Telecommunications with a difference!

Unexcelled communications power and compatibility, especially for professionals and serious computer users. Look us over; **SuperTerm** isn't just "another" terminal program. Like our famous Terminal-40, **it's the one others will be judged by.**

- **EMULATION**—Most popular terminal protocols: cursor addressing, clear, home, etc.
- **EDITING**—Full-screen editing of Receive Buffer
- **UP/DOWNLOAD FORMATS**—CBM, Xon-Xoff, ACK-NAK, CompuServe, etc.
- **FLEXIBILITY**—Select baud, duplex, parity, stopbits, etc. Even work off-line, then upload to system!
- **DISPLAY MODES**—40 column; 80/132 with side-scrolling
- **FUNCTION KEYS**—8 standard, 52 user-defined
- **BUFFERS**—Receive, Transmit, Program, and Screen
- **PRINTING**—Continuous printing with Smart ASCII interface and parallel printer; buffered printing otherwise
- **DISK SUPPORT**—Directory, Copy, Rename, Scratch

Options are selected by menus and EXEC file. Software on disk with special cartridge module. **Compatible with CBM and HES Automodems**; select ORIG/ANS mode, manual or autodial.

Write for the full story on SuperTerm; or, if you already want that difference, order today!

Requires: Commodore 64 or VIC-20, disk drive or Datasette, and compatible modem. VIC version requires 16K memory expansion. Please specify VIC or 64 when ordering.

Smart ASCII Plus . . . $59.95

The only interface which supports streaming —sending characters simultaneously to the screen and printer — with SuperTerm.

Also great for use with your own programs or most application programs, i.e., word processors. **Print modes:** CBM Graphics (w/many dot-addr printers), TRANSLATE, DaisyTRANSLATE, CBM/True ASCII, and PIPELINE.

Complete with printer cable and manual. On disk or cassette.

VIC 20 and Commodore 64 are trademarks of Commodore Electronics, Ltd.

(816) 333-7200

Send for a free brochure.

MIDWEST MICRO inc.

MAIL ORDER: Add $1.50 shipping and handling ($3.50 for C.O.D.); VISA/Mastercard accepted (card# and exp. date). MO residents add 5.625% sales tax. Foreign orders payable U.S.$, U.S. Bank ONLY; add $5 shp/hndlg.

311 WEST 72nd ST. • KANSAS CITY • MO • 64114

REVIEWS

(From p. 146)

Memory Lane

Realizing this, one of the first uses for microcomputers was to electronically organize information in the form of a database. In essence, a database manager collects and files all your records on disk so you can quickly and easily find information.

As micro systems became more powerful, so did database applications. Current programs can automatically perform calculations, sort information and generate reports containing just the information you need.

Yet even with all this power, there are trade-offs, usually involving the ease with which you can accomplish tasks. Take, for example, dBaseII (also by Ashton-Tate).

According to Datapro Research Corp. of Delran, NJ, dBaseII is the most widely used database program—it is also one of the most difficult to use. dBaseII is similar to a programming language, so it takes a lot of time and computer background to design an application using this program. In fact, dBaseII is used by programmers to write other programs.

Ashton-Tate's designers set out to find a way to keep the inherent power of dBaseII while making the program easy for even first-time computerists to use. The result is Friday!

A Solution

Friday! relies on a main menu with only three options (Fig. 1). These present you with several other choices from three additional menus. That's it—no layer upon layer of additional menus that make even a simple operation seem to take forever.

One of the problems with dBaseII is you have to remember a whole litany of commands and enter them using a precise syntax. There is no help function or other memory aid.

In contrast, Friday!'s simple menu structure is complemented with numerous well-placed prompts that guide you through each operation. These don't slow your progress because Friday! usually anticipates your response, so you simply hit the return key to answer most questions.

The first of Friday!'s three main menu selections concerns entering data. You use this option to choose a format for data you want to store, and then start entering data in the form you've established.

Electronic files that Friday! creates aren't much different from paper files. Perhaps you want to automate the records for some rental property you own. For each person, you keep a listing of name, address and specific information about the property, like amenities. You may also keep track of some math, like the commission due the salesperson renting the property.

Friday! lets you do all this automatically, without altering the basic flow of your manual system—except to make it more efficient. Friday! files are made up of individual records, such as all information related to a specific person; each record is then made up of various fields containing very specific data, such as that person's name.

The first step is to determine what information you want to store in each record. One caveat: it's best to do this basic layout on paper beforehand, because once you're committed to a design and start entering data, any changes to the design can destroy the previous records.

File Design

To get started designing a new file, select Enter Data from the main menu. A listing of all current files is displayed, along with other options, on the bottom of the screen. These additional choices permit you to create a new file, choose or edit an existing file or return to the main menu.

Since you're starting from scratch, select A-create to start a new file. A fresh screen is displayed with areas for you to enter the name, type and length of each field. Again, the bottom of the screen shows the available options: adding or inserting a new field, editing a current field or saving the form, among others.

Designing a data entry form is simply a matter of typing the name of each field, such as PHONE:NUM, and specifying the type and maximum number of characters the field can hold. Three type definitions are possible: anything, math and yes/no.

It's important to specify the proper definition at the outset, especially if the field will be part of a calculation. I found that if you don't take this precaution, Friday! won't let you specify this field in a formula, even if it contains a number.

A field holding alphanumeric data, like a name, is considered an anything field. Numeric data should be considered a math field, even if you aren't doing any calculations right away.

A yes/no field allows you to enter the characters Y or N. One possible use of this field is to decide whether a part in an inventory file should be reordered.

As you enter the field descriptions, a status line reminds you of how many fields are still available and the number of characters left.

Friday! permits a maximum of 32 fields in a record. The total length of all fields cannot exceed 999 characters. This is in keeping with the specifications of dBaseII, which was used to write Friday!

There were a few times where the 32-field limit made me compromise in designing a form: you must decide what information is crucial and leave less important fields off the layout. However, with some careful planning early on, you should be able to lessen the impact of this limitation.

At this point, you can save the form and start entering data. Each one of the fields you defined will be displayed on a separate line of the screen, with left and right brackets indicating where you can enter information. Just type the information in the proper spot, as shown in Fig. 2. When the entire form is filled in, the record will be automatically stored on disk.

Fab Features

Friday! has a number of features that make data entry accurate and easy to follow. You can ask Friday! to automatically compute any math field. Computed fields can be defined using the "other" main menu selection.

You can be shown a list of all the math fields you defined. Simply tell Friday! which field should hold the results of the calculation and enter the formula.

Another nice feature is that you can change the appearance of the data entry screen by placing fields anywhere on the screen and in any order. You can even create a customized entry form that contains only some of the fields from the original form. For instance, you may not want everyone using a certain form to see salaries, costs or other sensitive information.

The special form acts just like the original—you can view, edit and add records. You can still go back to using the original form at any time, with all the data still intact, because each entry form acts as a window into one data file. Each user may have a different perception of the file, but

```
                MAIN MENU
        A - Enter Data

        B - Retrieve Data

        C - Other Activities

        D - Leave
What do you want to do? [ ]

              RETRIEVE DATA
        A - View/Edit records
        B - Quick Report
        C - Custom Report
        D - Report Lookup
        E - Mailing Labels

        P - Main Menu
        M - Main Menu
What do you want to do?  [ ]
```

Fig. 1. Friday!'s main menu and submenu.

each is actually working with the same file.

Creating these special forms is a time-consuming process. You must type in the location (using row and column coordinates) where each field is to be placed. A guide on the screen displays column and row numbers to help you place fields in the right spot.

The easiest databases I've used (pfs:File and The Manager) let you design a form by typing field names anywhere you like. This makes the initial design much quicker, but doesn't permit the different views possible with Friday!, which requires a two-step definition process. In this case, what you sacrifice in ease-of-use is made up for with a lot more design flexibility.

Enter and Edit

After all your data is neatly organized in Friday!'s files, the next consideration is how to get at and change that data. Friday! has a number of easy ways to do this. While in the Data Entry mode, you can jump to a specific record number, or move forward and backward through the file one record at a time. When the record is displayed, data in any field can be modified.

This is fine if you know which record number holds the information you're looking for. But what happens if you don't? The retrieve data option allows you to use any field as a search field. Now, if you just know a person's last name—and LAST:NAME is the search field—simply entering the last name will find the record.

This process is quick, even with large files, because Friday! employs an efficient index scheme that tells it where records are located in the file.

This retrieve data option also permits you to sort the file using any field or to search through the file for all records that meet certain criteria. The search option has various levels of precision. If you specify M in the State field, Friday! retrieves all records containing states beginning with M. Likewise, typing "MISS" as a search criteria displays all records with Mississippi and Missouri.

As with the data entry option, once a record is displayed, you can modify it or even delete it. A fail-safe provision ensures that deleted records are not immediately erased from the disk; an Undelete command restores them to active status in case you have a change of heart.

Reporting Your Findings

The final major area Friday! covers is report generation, allowing you to print (or list on the screen) some or all of the information contained in your files.

Report generation follows the same type of philosophy used in form creation and data retrieval. Namely, you can create a quick report for simple listings of a file or more sophisticated customized reports.

In either case, establishing a report is like designing a customized data entry screen—you indicate the position on the screen where information should be placed. You can choose to print all or only some of the fields from the file.

```
File: EXAMPLES                ENTER DATA   View Record:    47                150


LAST:NAME   [Walker              ]
FIRST:NAME  [Rick          ]
ADDRESS     [345 East Academy Road    ]
CITY        [Provo               ]
STATE       [UT]
ZIP         [84605]
PHONE:NUM   [             ]
SALEPERSON  [JS]
TIME:AVAIL  [09/15/85]
BED:BATH    [05/03]
TYPE:UNIT   [Villa]
DRIVER      [N]
AMENITIES   [Pool, Tennis                    ]
RENT:MONTH  [   1000.00]
PERC:COMM   [ 0.071]
COMMISSION  [     71.00]


 A-Add          X-Delete         J-Jump           V-Video Off      >-Forward
 E-Edit         U-Undelete       F-File Menu      D-Ditto Off      <-Back
                C-Compute        L-Layout Menu    M-Main Menu      Choice   [<]
```

Fig. 2. Friday!'s data entry screen.

> Friday! is a powerful system that owes a lot to its dBaseII heritage.

Retrieval rules, like those used to find a particular record, are used to include only selected records in the report. Even with the quick report, you can have Friday! total or subtotal various fields.

Reports can be up to 160 characters wide, which is about the maximum that a daisy-wheel printer can print at 12 characters per inch. You also have a fair amount of control over the number of lines per page, titles and where page breaks occur. About the only feature missing is the ability to stop the printer after each page, which is needed if you use hand-fed single-sheet stationery.

I converted an invoicing program to Friday! for this evaluation. It worked fine until the printing stage. I only have access to individual multipart forms, and there was no way to stop printing while the next form was placed in the printer. Friday! let me down, but for most applications using continuous stationery, there is no need to worry—Friday! can accommodate your needs.

The custom report is very powerful. You can write a form letter that Friday! will later customize with information from the data file. An unusual feature is a lookup option that converts information stored in a record into another form. In the example shown in Fig. 2, only the salesman's initials are stored, rather than the full name.

For times when a full name is needed, the lookup function automatically converts the initials into a complete name, based on a table you create.

Friday! can also print mailing labels using any of the fields contained in the file (only five lines are available per label, but multiple fields can be placed on a line). You can place up to four labels across a page, with control over label width and lines between each label.

Newsworthy Notes

Friday! is a powerful system that owes a lot to its dBaseII heritage. The menus and prompts make it really simple to set up a data file and retrieve information. Friday!'s extensive use of error-checking is added insurance that you won't wipe out a 1000 record file with a single, mistaken keystroke.

Menus and prompts only accept valid options. You'll be asked to select again if an inappropriate choice is made. Whenever there is a chance of erasing or modifying a file with potential loss of data, several confirmations are required before the action really takes place.

Naturally, when you introduce simplicity into a program you run the risk of eliminating some of its power. That is true with Friday!, as noted with the rigid printer design that didn't let me use single sheets. However, it's easy to advance beyond Friday!

Friday! files can be used by dBaseII and vice versa. You might use Friday! to build screen layouts and simple reports that can form the core of a dBaseII application. Friday!'s report function for custom letters is a bit tedious. So you might consider using Friday! files with WordStar and MailMerge.

Friday! files can also be moved between Lotus's 1-2-3 integrated spreadsheet program for "what-if?" analysis of your data.

Friday!'s professional, typeset manual

is simple to follow. You start with the tutorial section, using sample files on a supplied data disk. When you get a bit more confident, start designing your own files according to the well-organized lessons.

Another unique feature of the program and documentation is that each display contains a reference number. By looking up that number in the manual's prompt section, you can immediately find a description of the operation and what you should be doing.

Friday! is well-supported by Ashton-Tate and its dealers. Post-sale support is included with the price of the package. Ashton-Tate's *dNews* monthly magazine contains Friday! programming tips, technical information and application stories.

Considering the easy way Friday! works, and the simple way data files can be transported to other applications and make up for the few limitations, you should give Friday! strong consideration when evaluating database applications.

Mike Heck
Harleysville, PA

Harvard Project Manager

System Requirements: IBM PC, XT or PC-compatible computer; 128Kb, two double-sided drives or a single double-sided drive and a fixed disk; a printer is helpful.

Manufacturer: Harvard Software, Inc., Harvard, MA 01460.

Price: $395.

When you do a lot of software reviews, it sometimes seems that the only software around is for word processing, spreadsheeting, database management or graphics. The variations on the theme are subtle, and after a while most of the packages start to look a lot alike.

When a program comes along that does something different or addresses a subject that's out of the ordinary, it's a real relief; so it was with a great deal of anticipation that I awaited the arrival of the Harvard Project Manager (HPM).

HPM is a project management tool. It allows you to apply Project Evaluation and Review Techniques (PERT) and the Critical Path Method (CPM) to projects. HPM also prepares Gantt charts on your project schedule, keeps track of tasks and milestones, monitors actual performance against anticipated performance and keeps track of expenses.

Before I go any further, I should explain a few terms for those not familiar with this style of project management.

A Few Terms . . .

PERT was developed to monitor projects that have many related and crucial pieces.

What is PERT? If you think of a project as a network of related tasks that lead to the accomplishment of a goal or the production of a product, you have a good grasp of what PERT is. With PERT, work is called a task and the product of those tasks is referred to as a milestone.

When a program comes along that does something different, it's a real relief; it was with a great deal of anticipation that I awaited the arrival of HPM.

There's an important distinction to keep in mind—a completed task requires the consumption of assets, money, time, materials and manpower; a milestone, since it is the product of the completed task, is an event that can be easily identified. It doesn't require the consumption of anything to be completed.

Many projects have more than one task going on at once. When building a house, the roof and exterior wall coverings can be going up at the same time. Some tasks take longer to complete than others. The ones that take the longest to complete are said to be on the critical path (CPM) because any delay in their completion slows down the entire project.

An activity not on the critical path is said to have slack time in it. That activity can be delayed for some amount of time without delaying completion of the entire project.

Documentation

The manual that comes with Harvard Project Manager is an interesting blend of tutorial and practice exercises. The first section of the manual gives a brief overview of what project planning is. This is followed by a series of disk demonstrations of the features of HPM.

After each demonstration, you're asked to try out the things you have learned. This is an effective way to get you quickly accustomed to the program.

The reference section is a model of brevity and conciseness. I was able to find the answers to almost all of my questions there, yet the entire section takes up only a few pages. The use of color printing to highlight text and the use of white space makes this one of the easiest-reading manuals I've seen in a long time.

The only detraction from this otherwise well-done manual is that the reference section may be too concise. Some information is hard to find. For example, information on printing a sorted list of the node data isn't in the manual.

Reference is made to the error window, and error messages do appear when something is amiss, but there is no list of error messages or the actions you can take to resolve the problems.

Program Particulars

When you load HPM, you're greeted with a screen divided into four windows (Fig. 3). In the upper left windows is a picture of the simplest of networks: a project with a start, a finish and a single task.

To the right is a picture of the schedule for the road map depicted in the left window. In the lower left is the calendar window, showing the calendar that has been selected for use with the project. Finally, the function window is in the lower right section.

The easiest way to explain how HPM works is to show how a project is scheduled using this product. Your project will be to build a house. You start by building a road map of the project.

When you select the road map function, the divided screen is replaced with a single screen, or window, into the project. This window contains the simple project shown in the small road map window in Fig. 3. You are going to start building your house in July and hope to have it completed and be moved into it before winter—say the first of November.

Setting Up

The first thing you need to do is set up a calendar to match the project work schedule. HPM allows you to designate holidays and workdays as well as the length of the workday. Assume a normal eight-hour day and a five-day workweek beginning on Monday and ending on Friday.

HPM is happy with this, but allows you to designate any schedule you want and also allows you to make changes to the workday and workweek at any time if you wish to see the effect on your project.

With the calendar set, call up the road map and schedule the foundation and framing of the house. To remind you of the date, briefly go into two windows so you can refer to the calendar as you are entering the first project information.

Each task and milestone on the network is called a node. As you add tasks and milestones to the network, you're prompted for information. At Fig. 4, you've just added the milestone "Poured". At the top of the window is the road map with the cursor highlighting the milestone.

The bottom window is the node window, which prompts for information about the project. Each node on the road map has an associated node window with information about projected and actual start times, cost, cost codes and person responsible.

The program also calculates and fills in some of the information on the node window based upon information you have already given the program. Entered

information is placed in highlighted boxes, while information computed by the program is displayed as bright characters on the screen.

Fig. 5 shows the other end of the road map. Notice that the number of working days to complete the project is displayed at the upper left of the screen while the upper right shows cost information. These numbers are automatically updated by the program as new information is entered. At the bottom of the screen is the help window.

Now that you have entered all of your costs and other data, you are ready to look at a schedule of your project. HPM can make what are commonly called Gantt charts. Solid bars represent the length of time it takes to complete each of the tasks; diamonds show milestone dates.

A Gantt chart can show that work on the roof, walls, plumbing and wiring is all going on at once; the chart can also depict which projects are on the critical path and must be completed before other work can continue.

As the house building project continues, you can use Harvard Project Manager to keep track of your actual performance compared to objectives. You'll be able to see very quickly if you are getting into trouble with your November 1 deadline.

As it turns out, the earliest estimated completion date is October 20 and the latest estimated completion date is October 27. With luck, help from HPM and good weather, you should be snug in your new home before winter sets in.

Printing and Reporting

While HPM's screen handling is excellent, the program would be of little use to a businessman if he couldn't get project data out of the program and onto paper.

The people at Harvard Software have excelled in making the print functions just as useful as the rest of the program. You can retrieve the node data in alphabetical order or sorted by either person responsible or cost code.

If your road map and schedule are too long to print on one sheet of paper, the program will either print segments of the function on separate sheets or, even more useful, print the entire function sideways on continuous sheets of paper. This is a much neater job than pasting together several sheets.

HPM supports some common printers, but the program could be beefed up to support more. The Epson MX- and FX-series of printers with Graphtrax Plus are supported, but the program doesn't take advantage of the wide carriage on the Epson 100-series printers.

If you don't wish to print your file immediately, it can be saved to a print file or you can save project data in a DIF (Data Interchange Format) files for integration into a spreadsheet.

A Bugless Landmark

Harvard Project Manager is a well-

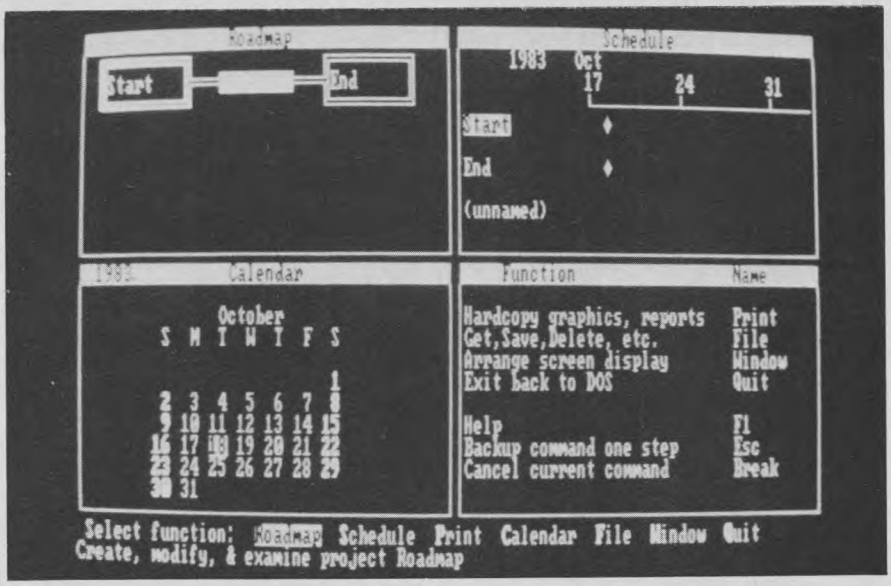


Fig. 3. The Harvard Project Manager screen, with windows.

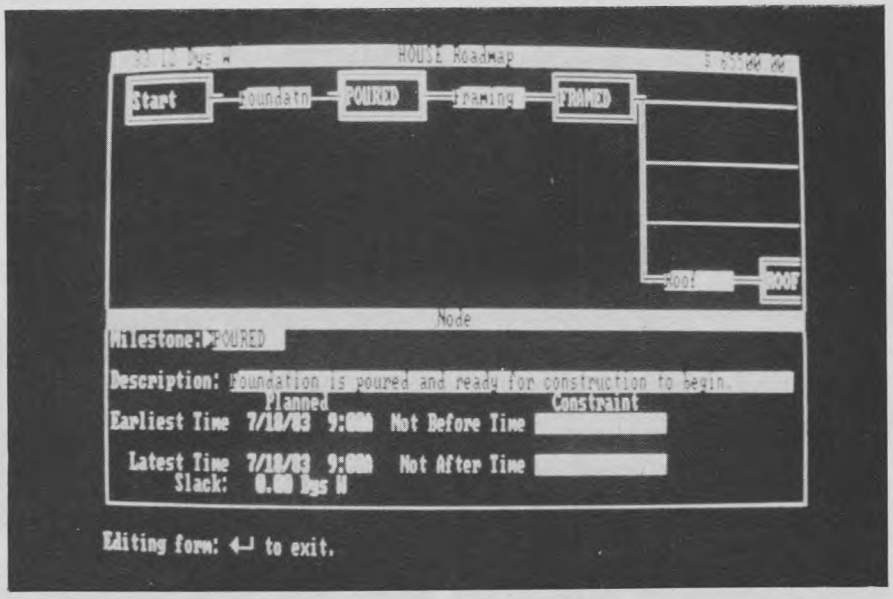


Fig. 4. The road map and milestone, top, and node window, bottom.

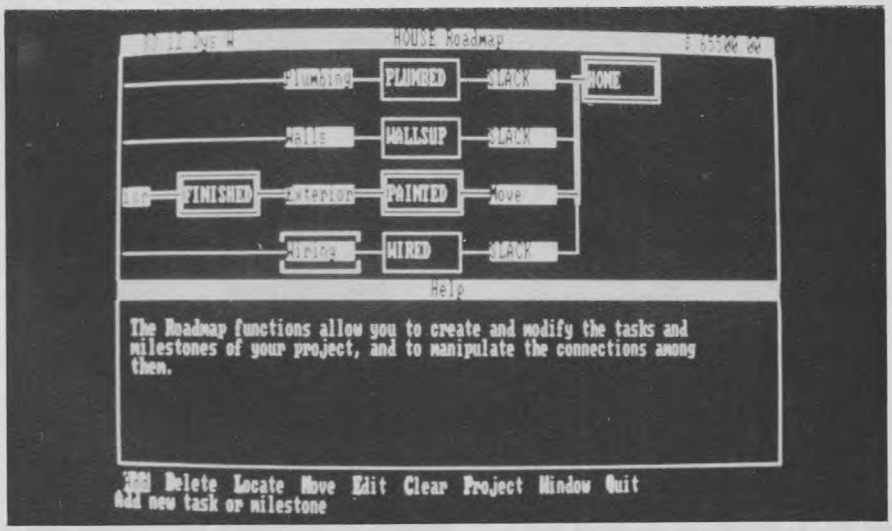


Fig. 5. The other end of the road map. Cost information is in the upper right corner.

designed program. The version I tested had no bugs in it that I could detect. I am sure that there are some there, waiting to be found, but I gave the program a good workout and was very pleased with the results.

One of the things that makes this a landmark program is the user interface. Harvard Software has taken great pains to provide users with a friendly and inviting work environment.

The use of windows is extremely well-done. Each window scrolls as you move about in the project. If you are in the road map window and also have the schedule window displayed, the schedule window will scroll vertically as you move horizontally on the road map.

HPM also takes advantage of the function keys available on the IBM keyboard. The home key will bring you to the beginning of a project; the end key moves you to the end of the project.

Tab keys are used to move you about in the node window as you are entering data. You can use the space bar to move the pointer to the function you wish to use on the menu, or you can press the first letter of the desired function to select it.

This program makes extensive use of graphics, but the graphics adapter is not required to run this program on the IBM PC. Harvard Software wisely chose to use the extensive set of characters that are available in the IBM PC to create the graphics.

The result is that this program can run on a PC equipped with either the IBM-supplied monochrome monitor or one equipped with the graphics adapter.

A Recommendation

The Harvard Project Manager is a hard-working management ally. Properly used, this program may well assist you in finding out how to put a little more profit in the bottom line by pointing out where your problems are. It is easy to use, comes with well-done documentation and seems to have support from its creators.

There are very few of us so well-organized that we can't profit from using this program. I recommend it highly to anyone using PERT and CPM now, or to anyone who has wanted to use these management methods but hasn't because of the effort required to draw charts and calculate times.

Shawn Bryan
Montpelier, VT

Uniform

System Requirements: Any Kaypro machine running CP/M.

Manufacturer: Micro Solutions, Inc., PO Box 15033, La Cruces, NM 88004.

Price: $100.

If ever a wedding between computer and software were ordained in Heaven, then the Kaypro 10 and the software program Uniform is definitely it.

Uniform answers one of the most pressing needs of the computer user—the need to read disks from different computer systems.

Until the arrival of Uniform, you needed a smart friend who could hardwire a cable and connect two different systems, and both usually had to be in the same room. Or, you needed a modem and a communications program to send programs or data over the telephone line to someone else. Of course, file transfers are much slower over the telephone line. In some cases it just isn't practical for you to box up all of your computer equipment and go over to a friend's to share programs you've worked up.

Now It's Simple

With Uniform, designed to work on the

Single-Sided Formats

Kaypro
Osborne 1 (SD)
Osborne 1 (DD)
Xerox 820 (SD)
Xerox 820-II (DD)
TRS-80 Model I (Omikron CP/M)
TRS-80 Model III (MM CP/M)
IBM PC (CP/M-86 SS/DD)
Morrow MD2
NEC PC-8001A
Zenith Z90 (48 tpi DD)
Heath with Magnolia (DD)
Superbrain (JR)
Superbrain (40 trk SS)
Cromemco CDOS (SSSD)
Cromemco CDOS (SSDD)
Cromemco with Intl. Term. CP/M
DEC VT-180 (DD)
Access (SS/DD)
Lobo Max-80 (DD)
TI Professional (DD)

Double-Sided Formats

Kaypro
Televideo TS-802
Hewlett-Packard HP-125
Otrona Attache
IMS 5000
Epson QX-10
Sanyo
IBM PC (CP/M-86 DS/DD)
Morrow MD3
NEC PC-8801A
Zenith Z100 (DD)
Datavue
Superbrain (QD)
Magic Computer
Cromemco CDOS (DS/SD)
Cromemco CDOS (DS/DD)
Cromemco with Intl. Term. CP/M

Fig. 6. SETDISK.COM reads and writes to 38 computer disk formats.

popular Kaypro transportable computers, you now need to simply pop your friend's disk into your Kaypro to have a copy of his program, which can now run on your Kaypro. The Uniform program I tested was designed to operate on the hard disk-equipped Kaypro 10.

Uniform by Micro Solutions
version 2.10 [00 serial no.] for Kaypro 10

1...SETDISK
2...INITDISK
3...MSDOS/PCDOS ¼-½ CP/M file copy
4...TRS DOS/LDOS -½ CP/M file copy

Which program would you like (or RETURN to quit?)

Fig. 7. Uniform's main menu.

Uniform, as provided for the Kaypro 10, consists of two programs: UNIFORM.COM is an 8Kb command file; UNIFORM.UVL is a 64Kb program.

The prototype version of Uniform, designed to run on the Kaypro II, consisted of two programs: SETDISK.COM and INITDISK.COM. Uniform version 2.10 for the Kaypro 10 incorporates both of these programs in UNIFORM.COM and adds two other programs. One of the extra programs copies TRS DOS or LDOS files to CP/M and vice versa. The other copies MS DOS or PC DOS to CP/M and, again, vice versa.

SETDISK.COM has been expanded in its Kaypro 10 version to let you set disk drive C to read and write to 38 different computer disk formats (see Fig. 6). Twenty-one of these are single-sided formats; the rest are doubled-sided formats.

Using Uniform on the Kaypro 10 is as simple as typing UNIFORM and pressing return. This puts the menu shown in Fig. 7 on the screen.

To use Uniform, first decide which program you want to use. For example, if you have borrowed a disk from a friend who owns a Superbrain (which uses single-sided, 40-track disks), you need to run SETDISK.COM. Setdisk lets you set drive C on the Kaypro 10 to pretend it is a Superbrain drive. You can then use CP/M's PIP to transfer write files between the two formats. Uniform can also format disk in the 41 formats listed in Fig. 8.

Single-Sided Formats

Kaypro II
Osborne 1 (SD)
Osborne 1 (DD)
Xerox 820 (SD)
Xerox 820-II (DD)
TRS-80 Model I (Omikron CP/M)
TRS-80 Model III (MM CP/M)
IBM PC (CP/M-86 SS/DD)
Morrow MD2
NEC PC-8001A
Zenith Z90 (48 tpi SS)
Heath with Magnolia (DD)
Superbrain (JR)
Superbrain (40 trk SS)
Cromemco CDOS (SS/SD)
Cromemco CDOS (SS/DD)
Cromemco with Intl. Term. CPM
DEC VT-180 (DD)
Access (SS DD)
Lobo Max-80 (DD)
TI Professional (DD)
MS DOS/PC DOS 1.x

Double-Sided Formats

Kaypro 4/10
Televideo TS-802
Hewlett-Packard HP-125
Otrona Attache
IMS 5000
Epson QX-10
Sanyo
IBM PC (CP/M-86 DS/DD)
Morrow MD3
NEC PC-8801A
Zenith Z100 (DD)
Datavue (DS/DD)
Superbrain (QD)
MAGIC Computer
Cromemco CDOS (DS/SD)
Cromemco CDOS (DS/DD)
Cromemco with Intl. Term. CPM
MS DOS/PC DOS 1.x
MS DOS/PC DOS 2.x

Fig. 8. Uniform can format 41 disks.

Documentation

The Uniform User's Guide is a 14-page, staple-bound manual with a slick cardboard cover. It has a table of contents but no index. Given the simplicity of the program, however, the omission of an index with a manual of this size won't really slow up anyone. The table of contents is adequate to provide help in locating specific information.

Performance

The copy programs work as noted in the manual. I tried copying a TRS-80 Model III disk and had no problems with the transfer from the Model III disk to the Kaypro's hard disk. Just as the Uniform manual notes, the TRS DOS programs I copied did not work on the Kaypro. (On p. 13, the Uniform manual warns, "TRS DOS/LDOS programs will not work on a CP/M computer. This program will *only* copy ASCII text files properly.")

The Uniform program makes my computer complete. It's hard to believe that computer manufacturers have not agreed on a standard 5¼-inch disk format. The developers of Uniform have done a service to the computer industry by devising a way to use disks of various formats.

Bob Hickey
Eagle River, AK

Scrutinizing Wall Street
A Landmark Project Manager
Keep Kaypro Disks on Speaking Terms
dBasell's Powerful Little Brother

Personal Investment Analysis

System Requirements: IBM PC or compatible; 64Kb; BasicA; one disk drive.

Manufacturer: Wiley Professional Software, 605 Third Ave., New York, NY 10158.

Price: $60.

Personal Investment Analysis, from Wiley Professional Software, is a package of dynamite for the individual who is interested in a variety of investment situations. It's good, it's powerful and it's capable.

First, the Bad

I'll quickly dispense with the negative aspects of the software. For all its capability, the package lacks operational sophistication. To begin with, you are emphatically cautioned that the program is protected under copyright, only to discover that you can list the program (it's in Basic). It seems that Wiley would have been wise to protect its own investment by storing the program in a protected manner, which isn't difficult to do.

The second major omission is in the inability to print the output in other than a screen print. While the program does a good job of producing amortization schedules, for example, there is no facility to provide them as output to a printer. This program, which would otherwise be extremely helpful to any realtor, is limited by a lack of sophistication.

The application is straightforward—no escape mechanisms. To return to the main menu you must complete the application you have selected. This is a minor disadvantage, but a major annoyance for someone who is accustomed to more forgiving software.

This concern is also prevalent when you see the mixture of response. You cannot, for example, respond to yes/no questions without typing the complete word when a Y or N would be sufficient. One-character responses require the use of the enter key in some instances and then avoids this requirement in others. The program also lacks simple screen management techniques.

Lest you be concerned that the package is for the IBM PC only, I was able to use it on some PC compatibles, specifically the Compaq and the Eagle.

Now, the Good

Once you get beyond the operational characteristics, however, the software is perfectly functional in performing analyses on these areas of personal finance: convertible bonds, home mortgages, interest rates, retirement funding and tax-free securities.

- Convertible bond analysis produces information on bonds that pay an annual dividend. While the program provides a selection of compounding or maturation dates for other options, this particular application is limited to annual calculations. If your bonds mature at different intervals, you'll have to make adjustments to the figures.
- Home mortgage analysis calculated my mortgage to the penny. This section provides periodic payment and amortization schedules for home mortgages. Balloon payments can be considered and the schedule partitions to accommodate such payments.

However, the results may not be totally meaningful—perfectly reasonable input may produce negative output. Since negative mortgages don't exist, the data is open to interpretation. The documentation, which is otherwise of high quality, isn't very clear on this point.

- The interest rate calculator determines the effective yield on investments that produce fixed annual incomes. Analysis of other than annual yields is not provided. Here again the program is inconsistent—a selection of yields would certainly be useful.
- Retirement funding projects the growth of savings given certain known factors. You should keep in mind that the program isn't clairvoyant and cannot gauge possible changes in the money market. The greatest strength of this portion of the program is its ability to project according to experience or to retroject from a stated figure.
- Tax-free securities determine the interest rate that must be earned to equal the earned interest on a tax-free security. It's good.

The documentation is sufficient for even a neophyte. I do recommend that Wiley modify its setup procedure to accommodate the differences in the Basics used by the Compaq and other IBM-compatible systems. Overall, the price is reasonable and the advice is competent.

Ken Lord
Winchendon, MA

Correction: In January's Software Review section, Infotory, from SSR Corporation, should have been priced at $425 for the floppy disk version and $575 for the hard disk version.

Friday!

System Requirements: Any computer running CP/M or MS DOS; 64Kb (128Kb for MS DOS; 56Kb for Apple III); dual floppy drives with at least 126Kb; an 80-column terminal or display.

Manufacturer: Ashton-Tate, 10150 W. Jefferson Blvd., Culver City, CA 90230.

Price: $295.

Whatever business or activity you're involved in, chances are information plays a vital role in most areas of your operation. The more organized and accurate the information, the better a chance you have for success.

(Continued on p. 140)

Introducing the Most Powerful Business Software Ever!

TRS-80™ (Model I, II, III, or 16) • APPLE™ • IBM™ • OSBORNE™ • CP/M™ • XEROX™

The VERSABUSINESS™ Series

Each **VERSABUSINESS** module can be purchased and used independently, or can be linked in any combination to form a complete, coordinated **business system.**

VERSARECEIVABLES™ $99.95

VERSARECEIVABLES™ is a complete menu-driven accounts receivable, invoicing, and monthly statement-generating system. It keeps track of all information related to who owes you or your company money, and can provide automatic billing for past due accounts. VERSARECEIVABLES™ prints all necessary statements, invoices, and summary reports and can be linked with VERSALEDGER II™ and VERSAINVENTORY™.

VERSAPAYABLES™ $99.95

VERSAPAYABLES™ is designed to keep track of current and aged payables, keeping you in touch with all information regarding how much money your company owes, and to whom. VERSAPAYABLES™ maintains a complete record on each vendor, prints checks, check registers, vouchers, transaction reports, aged payables reports, vendor reports, and more. With VERSAPAYABLES™, you can even let your computer automatically select which vouchers are to be paid.

VERSAPAYROLL™ $99.95

VERSAPAYROLL™ is a powerful and sophisticated, but easy to use payroll system that keeps track of all government-required payroll information. Complete employee records are maintained, and all necessary payroll calculations are performed automatically, with totals displayed on screen for operator approval. A payroll can be run totally, automatically, or the operator can intervene to prevent a check from being printed, or to alter information on it. If desired, totals may be posted to the VERSALEDGER II™ system.

VERSAINVENTORY™ $99.95

VERSAINVENTORY™ is a complete inventory control system that gives you instant access to data on any item. VERSAINVENTORY™ keeps track of all information related to what items are in stock, out of stock, on backorder, etc., stores sales and pricing data, alerts you when an item falls below a preset reorder point, and allows you to enter and print invoices directly or to link with the VERSARECEIVABLES™ system. VERSAINVENTORY™ prints all needed inventory listings, reports of items below reorder point, inventory value reports, period and year-to-date sales reports, price lists, inventory checklists, etc.

VERSALEDGER II™ $149.95

VERSALEDGER II™ is a complete accounting system that grows as your business grows. VERSALEDGER II™ can be used as a simple personal checkbook register, expanded to a small business bookkeeping system or developed into a large corporate general ledger system **without any additional software.**

- VERSALEDGER II™ gives you almost unlimited storage capacity (300 to 10,000 entries per month, depending on the system),
- stores all check and general ledger information forever,
- prints tractor-feed checks,
- handles multiple checkbooks and general ledgers,
- prints 17 customized accounting reports including check registers, balance sheets, income statements, transaction reports, account listings, etc.

VERSALEDGER II™ comes with a professionally-written 160 page manual designed for first-time users. The VERSALEDGER II™ manual will help you become quickly familiar with VERSALEDGER II™, using complete sample data files supplied on diskette and more than 50 pages of sample printouts.

SATISFACTION GUARANTEED!

Every **VERSABUSINESS™** module is guaranteed to outperform all other competitive systems, and at a fraction of their cost. If you are not satisfied with any **VERSABUSINESS™** module, you may return it within 30 days for a refund. Manuals for any **VERSABUSINESS™** module may be purchased for $25 each, credited toward a later purchase of that module.

To Order:

Write or call Toll-free (800) 431-2818
(N.Y.S. residents call 914-425-1535)

* add $3 for shipping in UPS areas
* add $4 for C.O.D. or non-UPS areas
* add $5 to CANADA or MEXICO
* add proper postage elsewhere

DEALER INQUIRIES WELCOME

All prices and specifications subject to change / Delivery subject to availability.

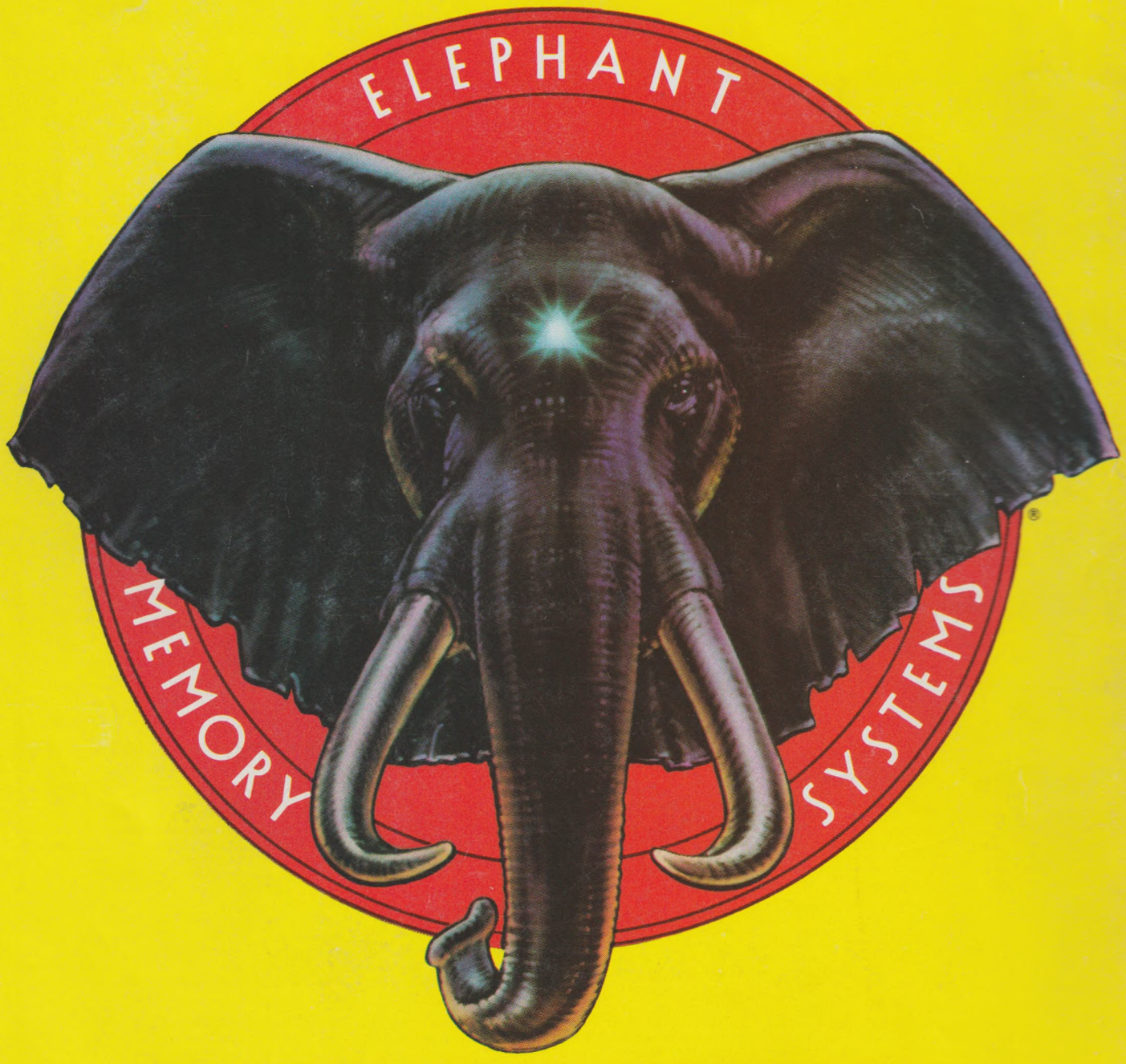
THE COMPUTRONICS INC

50 N. PASCACK ROAD, SPRING VALLEY, N.Y. 10977

* TRS-80 is a trademark of the Radio Shack Division of Tandy Corp. · *APPLE is a trademark of Apple Corp. · *IBM is a trademark of IBM Corp. · *OSBORNE is a trademark of Osborne Corp. *CP/M is a trademark of Digital Research · *XEROX is a trademark of Xerox Corp.

Circle 6 on Reader Service card.

REMEMBER.



ELEPHANT™ NEVER FORGETS.

A full line of top-quality floppies, in virtually every 5¼" and 8" model, for compatibility with virtually every computer on the market. Guaranteed to meet or exceed every industry standard, certified 100% error-free and problem-free, and to maintain its quality for at least 12 million passes (or over a lifetime of heavy-duty use).

Contact Dennison Computer Supplies, Inc., 55 Providence Highway, Norwood, MA 02062 or call toll-free 1-800-343-8413. In Massachusetts, call collect (617) 769-8150. Telex 951-624.

Dennison

Circle 355 on Reader Service card.